

AD-A046 348

ARGONNE NATIONAL LAB ILL  
AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)  
APR 77 D J BINGAMAN

F/G 13/2

UNCLASSIFIED

CEEDO-TR-76-34

NL

1 OF 3

AD  
A046348



AD A 0 46348



**CEEDO**



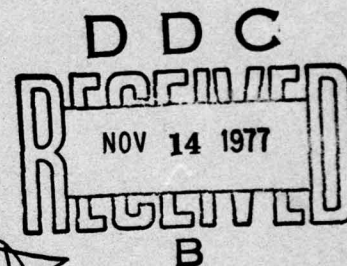
**CEEDO-TR-76-34**

**AIR QUALITY ASSESSMENT MODEL FOR  
AIR FORCE OPERATIONS—SHORT-TERM  
EMISSION/DISPERSION COMPUTER  
CODE DOCUMENTATION**

**ARGONNE NATIONAL LABORATORY  
9700 SOUTH CASS AVENUE  
ARGONNE, ILLINOIS 60439**

**APRIL 1977**

**FINAL REPORT FOR PERIOD  
1 JULY 1975-1 JANUARY 1977**



Approved for public release; distribution unlimited

**CIVIL AND ENVIRONMENTAL  
ENGINEERING DEVELOPMENT OFFICE**

**(AIR FORCE SYSTEMS COMMAND)**

**TYNDALL AIR FORCE BASE  
FLORIDA 32403**

**AJ No. \_\_\_\_\_  
DDC FILE COPY**



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CEEDO-TR-76-34	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-TERM EMISSION/DISPERSION COMPUTER CODE DOCUMENTATION	5. TYPE OF REPORT & PERIOD COVERED Final Report 1 Jul 1975 to 1 Jan 1977	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Dorothy J. Bingham	8. CONTRACT OR GRANT NUMBER(s) Project Order 76-0003	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Argonne National Laboratory 9700 South Cass Avenue Argonne IL 60439	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62601F/1900/5A03	
11. CONTROLLING OFFICE NAME AND ADDRESS Civil Environmental & Engineering Development Office Det 1 HQ ADTC Tyndall AFB FL 32403	12. REPORT DATE Apr 1977	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) 12 206p.	13. NUMBER OF PAGES 209	
	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. Ed Mr Smith		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) 9/3/2 739 7711 73026 B		
18. SUPPLEMENTARY NOTES 3029 8/970 2996 Available in DDC.		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Aircraft Assessment Airport Models Air Pollution Dispersion Model Computer Code 872-226 4013		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Air Force contracted with Argonne National Laboratory to develop a series of computer programs called the Air Quality Assessment Model (AQAM). The source emissions inventory routine of AQAM was designed to handle complex emission sources with emphasis on aircraft. A short term emission/dispersion model for hourly air quality predictions and a long term emission/dispersion model for monthly and annual predictions are also in AQAM. This report documents only the short term model. Flow charts, computer		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

033550 SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Block 20 continued

listings, and brief descriptions of each subroutine are included. They are intended for readers with a computer background who wish to examine or alter the computer code.

ACCESSION for	
NTIS	Write Section <input checked="" type="checkbox"/>
DDC	Write Section <input type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DIS	/or SPECIAL
A	

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

# PREFACE

This report documents work performed during the period 1 July 1975 through December 1976 by Argonne National Laboratory. The technical work for this effort was performed under the auspices of the Air Force Civil Engineering Center (AFSC) which on 8 April 1977, reorganized into Detachment 1 (CEEDO) HQ ADTC, Tyndall Air Force Base, Florida, 32403. Captain Dennis F. Naugle, CEEDO/ECA, managed the program.

This report has been reviewed by the Information Officer and is releasable to the National Technical Information Service (NTIS). At NTIS it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

*Dennis F. Naugle*

DENNIS F. NAUGLE, Capt, USAF, BSC  
Chief, Environmental Modeling  
Branch

*Joseph S. Pizzuto*  
JOSEPH S. PIZZUTO, Col, USAF, BSC  
Commander

*Pt S. Daley*

PETER S. DALEY, Maj, USAF, BSC  
Chief, Environmental Assessment  
Research Division

*Peter A. Crowley*  
PETER A. CROWLEY, Maj, USAF, BSC  
Director of Environics



# TABLE OF CONTENTS

	<u>Page</u>
INTRODUCTION	1
Subroutine ABABAR	6
Subroutine ABLNAR	14
Subroutine ABPTAR	19
Subroutine ACSRCE	24
Subroutine AINE	39
BLOCK DATA	43
Subroutine CAVL	45
Subroutine CLASSE	65
Subroutine DEPART	68
Subroutine DIFERF	72
Subroutine EMISAR	75
Subroutine ENARAY	78
Subroutine INDINP/DEPINP	83
Program MAIN	90
Subroutine MAINS	98
Subroutine METHA	106
Subroutine METHB	111
Subroutine METHC	114
Subroutine METHD	117
Subroutine METHE	120
Subroutine OUTPUT	123
Subroutine PLRISE	128
Subroutine POLSOR	135
Subroutine PSEUDO	146
Subroutine QMOD	149
Subroutine READ	153
Subroutine RISE	158
Subroutine RRDIST	162
Subroutine SIGY/SIGCY	167
Subroutine SIGZ/SIGCZ	170
Subroutine SOURCE	175

# TABLE OF CONTENTS (CONCLUDED)

	<u>Page</u>
Subroutine STPOL1	178
Subroutine STPOL2	185
Subroutine TRAN	190
REFERENCES	203

## SECTION I INTRODUCTION

Argonne National Laboratory (ANL) has developed an Air Quality Assessment Model (AQAM) for airbase operations under contract to the U.S. Air Force Civil Engineering Center (AFCEC) designed to simulate the emission of pollutants from sources on an airbase and the dispersion of these emissions in the atmosphere so as to enable calculation of pollutant concentrations over a grid of ground level receptors. These models are comprised of four physically separate computer codes, of which three must be operated by the user. The fourth code prepares a magnetic tape containing long term stability-time-wind roses for use by the long term climatological type air pollution model. This code is operated on request by the USAF Environmental Technical Applications Center in Washington, D.C. and the resultant magnetic tapes containing the climatological information is shipped to the user. The other three codes, developed by ANL, consist of the

- Source Inventory Model (SRCINV)
- Short Term Emission/Dispersion Model
- Long Term Emission/Dispersion Model

This report constitutes the computer code documentation for the second of these - the Short Term Emission/Dispersion Model. A separate computer code documentation manual (Reference 1) is available for SRCINV. Documentation for the Long Term Emission/Dispersion Model is currently being prepared and will be available shortly. A companion document to these reports - Operator's Guide (Reference 2) of the Air Quality Assessment Model for airbase operations - consists of a detailed discussion of the various functional parts of the computer programs and the input/output requirements. A second companion report (Reference 3) discusses the technical and theoretical basis underlying AQAM and presents and describes equations and algorithms used in the various AQAM sub-models.

The intended purpose of the present document is to provide a computer programmer with sufficient information so that he can study the code and make changes or modifications to it where required.



Table 1 contains a list of all routines contained in the Short Term Model in alphabetical order together with a brief description. More detailed descriptions of each routine, together with flow charts and computer code listings with comments that are intended to link listings to flow charts, are given on subsequent pages. It is hoped that this information, when combined with that given in References 1, 2, and 3, will enable a programmer to understand and make changes to the code when desired.



TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS  
IN THE SHORT TERM EMISSION/DISPERSION MODEL

<u>SUBROUTINE</u>	<u>DESCRIPTION</u>
ABARAR	Input airbase non-aircraft area source data from master source tape and compute emissions rates.
ABLNAR	Input airbase non-aircraft line source data from master source tape and compute emission rates.
ABPTAR	Input airbase non-aircraft point source data from master source tape and compute emission rates.
ACSRCE	Set up the aircraft source arrays and allocate emissions to areas and/or lines.
AINE	Translate line and receptor coordinates and set all necessary line parameters.
BLOCK DATA	Initialize data in common blocks.
CAVL	Compute coupling coefficient at a receptor due to a line source.
CLASSE	Print input error message.
DEPART	Calculate points in the departure path.
DIFERF	Find the difference between two error functions.
EMISAR	Accumulate emissions from airbase areas and lines.
ENARAY	Input environ source data from master source tape and compute emission rates.
INDINP/DEPINP	Print the source input.
MAIN	Read general data and direct control to READ and MAINS.
MAINS	Main driver for short term model.
METHA- METHE	Calculate diurnal emissions from non-aircraft sources using varying methods.
OUTPUT	Print pollutant concentrations at all receptors.
PLRISE	Calculate effective height and dispersion coefficients for a stack plume.
POLSOR	Direct calls to the proper diffusion routine for all input sources.



TABLE 1. LIST OF ALL PROGRAMS AND SUBPROGRAMS IN  
THE SHORT TERM EMISSION/DISPERSION MODEL (CONCLUDED)

PSEUDO	Call functions to find virtual distance from source to pseudo upwind point.
QMOD	Compute linear distribution of pollution along a runway.
READ	Read master source tape.
RISE	Calculate plume rise.
RRDIST	Calculate length of runway necessary for takeoff.
SIGY/SIGCY	Calculate horizontal dispersion or corresponding virtual distance.
SIGZ/SIGCZ	Calculate vertical dispersion or corresponding virtual distance.
SOURCE	Driver for non-aircraft emission routines.
STPOL1	Determine pollutant concentrations from point and area sources.
STPOL2	Determine pollutant concentrations from line sources.
TRAN	Calculate the coupling coefficient at a receptor due to a point or area source.

## SUBROUTINE ABARAR

### Purpose:

1. To read from the master source type all data needed to define airbase non-aircraft area sources.
2. To compute the emission rates due to evaporative hydrocarbons, space heating, off-road vehicles, and military and civilian vehicles.

### Input:

If the diurnal distribution cards are input, an additional parameter, IOPT, is read here to choose the method of distribution of those evaporative hydrocarbons not using the default of a uniform distribution.

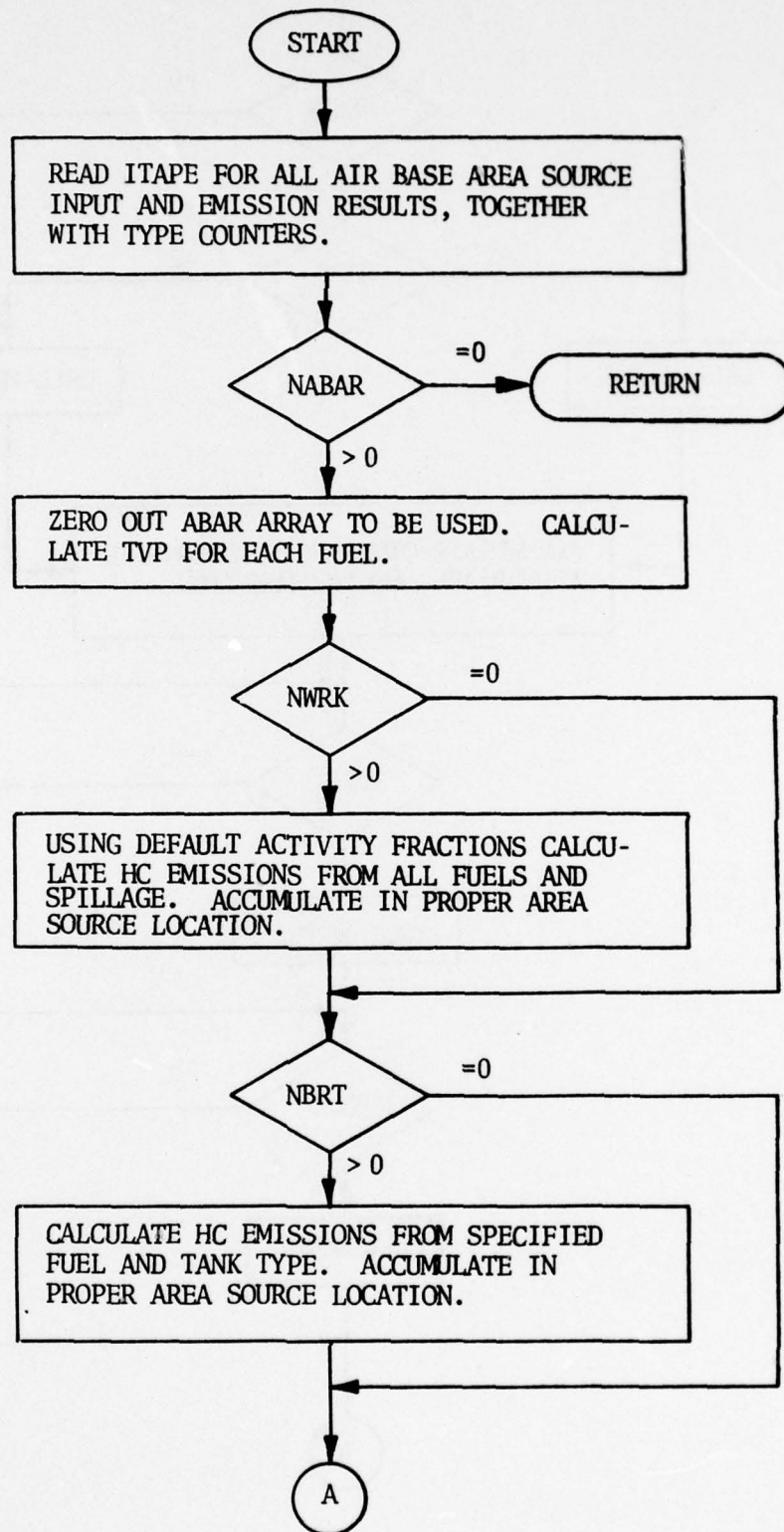
### Output:

The array, ABAR, is filled with geometry and emission data for airbase non-aircraft area sources.

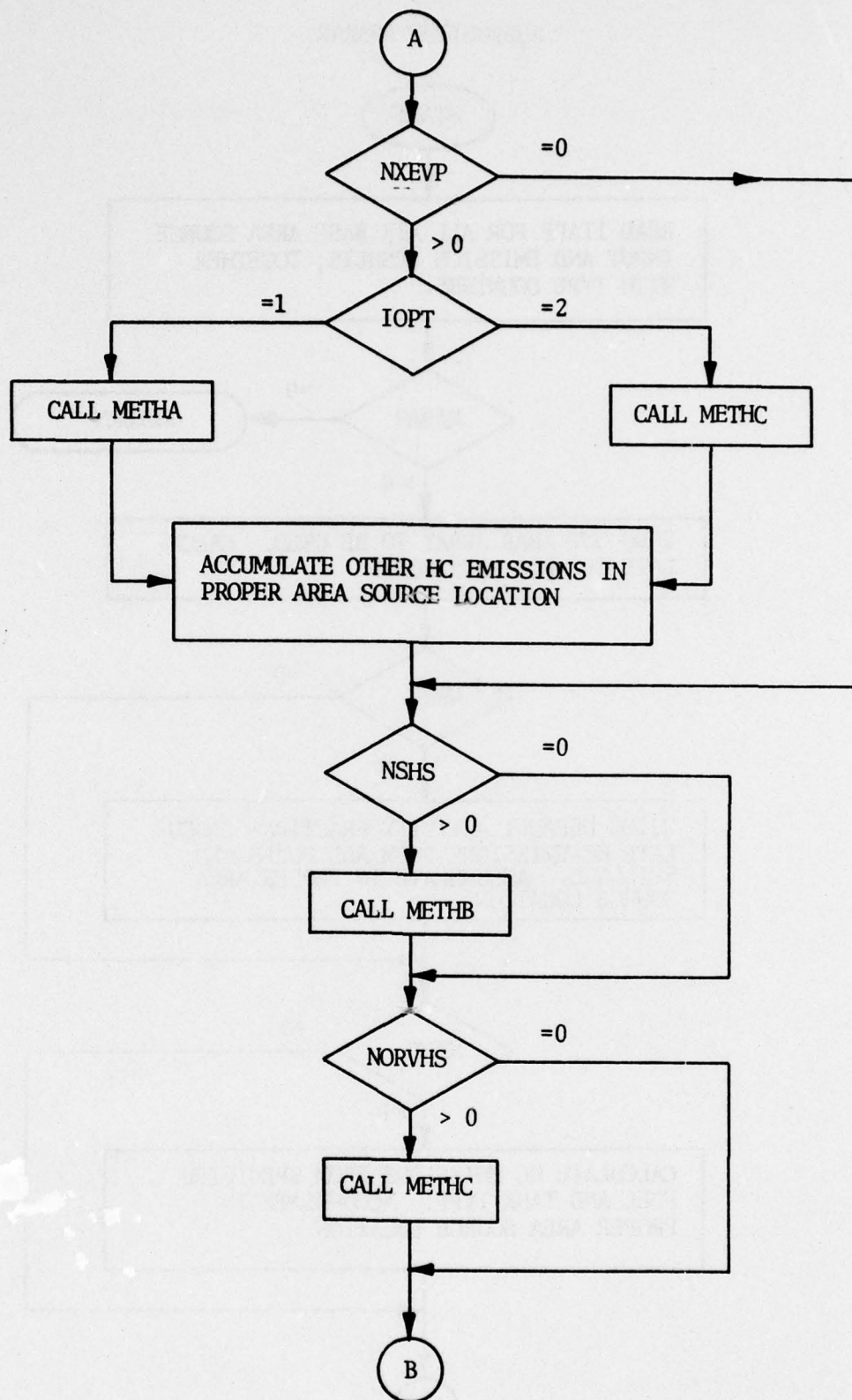
### Subroutines Called:

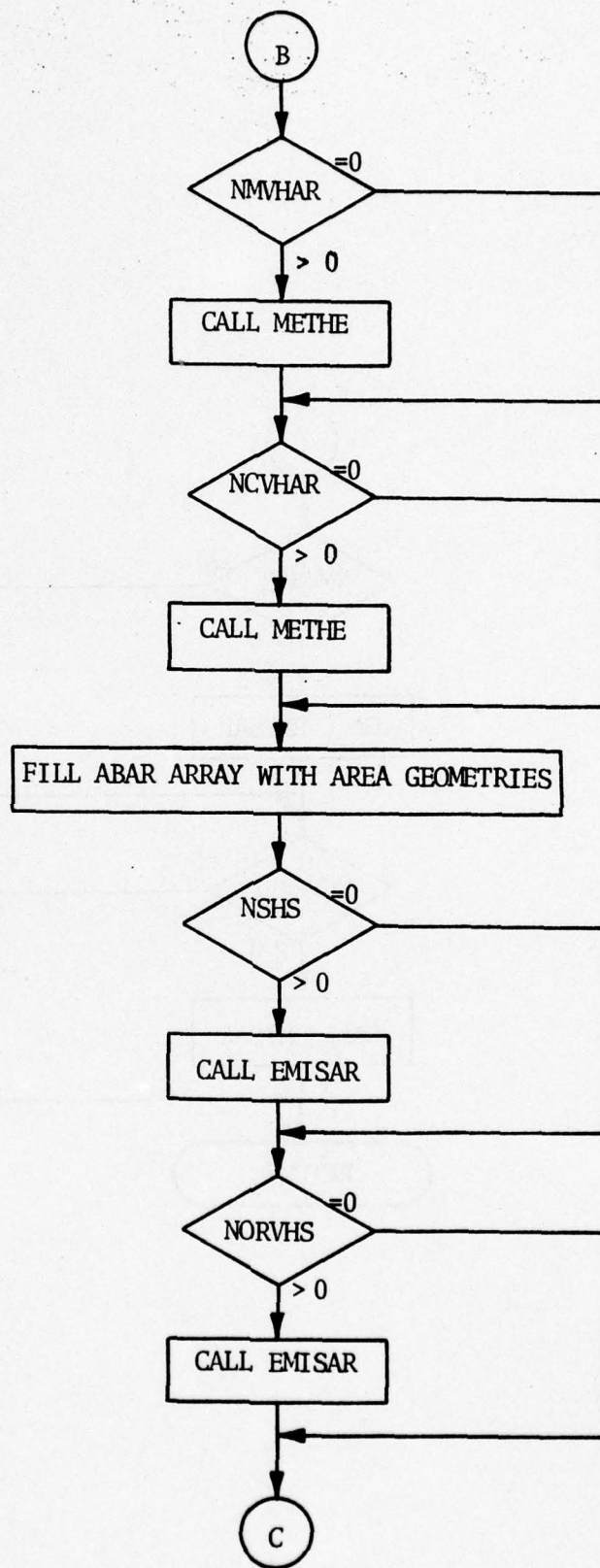
METHA, METHB, METHC, METHD, EMISAR

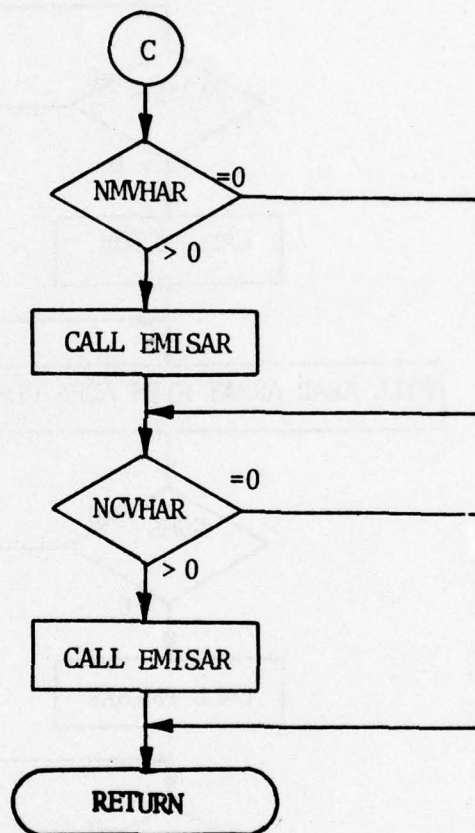
SUBROUTINE ABARAR













C	SUBFCUINE ABARAR	ABARR000
C		ABARR001
C	THIS ROUTINE COMPUTES THE EMISSION RATES FOR	ABARR002
C	ALL AIRBASE AREAS	ABARR003
C	NWRK = NO. OF HYDROCARBON WORKING LOSSES	ABARR004
C	NEFT = NO. OF HYDROCARBON BREATHING LOSSES	ABARR005
C	NXEVP = NO. OF OTHER EVAPORATIVE HYDROCARBON SOURCES	ABARR006
C	NSHS = NO. OF SPACE HEATING SOURCES	ABARR007
C	NCRVHS = NO. OF OFF-ROAD VEHICLE SOURCES	ABARR008
C	NMVHAR = NO. OF MILITARY VEHICLE AREA SOURCES	ABARR009
C	NCVHAR = NO. OF CIVILIAN VEHICLE AREA SOURCES	ABARR010
C		ABARR011
	CCMMCN /PERIOD/ IMONTH,NODAYS, IDAY, IHR1, IHR2, IFLAG, JFLAG	ABARR012
	CCMMCN / DEFAULT / ITAPE, ACLNDY, ACLNDZ, ALPHA (7), BETA (7), FLDENS (7)	ABARR013
	COMMON /DSTRBT/ ACMO (13,8), ACDY (2,8), ACHR (24,8), VHMLMO (13),	ABARR014
	. VHMLDY (2), VHMLHR (24), CVABMO (13), CVAEDY (2), CVABHR (24), CVENMO (13),	ABARR015
	. CVENDY (2), CVENHR (24), FLMO (13,7), FLDY (2,7), FLHR (24,7), NC1	ABARR016
	CCMMCN/JUNK/DAYS, LSRCE, NSRCE, SORCE (17,300), SORGH (10,200)	ABARR017
	. ,LCC1, LOC2, NGECM, IFT	ABARR018
	COMMON/MONMET/TMBAR, WSMBAR, AMDMBR, DTMBAR	ABARR019
	CCMMCN /SRCE/ NPLTS, NENPT, NENAR, NENLN, NABPT, NABAR, NABLN,	ABARR020
	. NACPT, NACAR, NACLN, ENET (16,100), ENAR (11,100), ENLN (14,20),	ABARR021
	. ABET (16,150), ABAR (11,100), ABLN (14,100)	ABARR022
	DIMENSION ABARGM (7,100), HCWRK (10,50), HCBRT (5,100), HCEVP (3,50),	ABARR023
	. FLHCUF (7), TVP (7)	ABARR024
	EQUIVALENCE (SORGM (1), ABARGM (1)), (SORGM (701), HCWRK (1)),	ABARR025
	. (SORGM (1201), HCBRT (1)), (SORGM (1701), HCEVP (1))	ABARR026
	LCC1=2	ABARR027
	LCC2=2	ABARR028
	NGECM=0	ABARR029
	IFT=0	ABARR030
	NSRCE=0	ABARR031
	I1=17	ABARR032
	I2=300	ABARR033
C		ABARR034
	READ (ITAPE) NABAR, NTOT, NWRK, NBRT, NXEVP, NSHS, NCRVHS,	ABARR035
	. NMVHAR, NCVHAR, NABARS, ((ABARGM (I,N), I=1,7), N=1, NABAR),	ABARR036
	. ((HCWRK (I,N), I=1,10), N=1, NWRK),	ABARR037
	. ((HCBRT (I,N), I=1,5), N=1, NBRT),	ABARR038
	. ((HCEVP (I,N), I=1,3), N=1, NXEVP),	ABARR039
	. ((SORCE (I,N), I=1, NTCT), N=1, NABARS)	ABARR040
C		ABARR041
	IF (NABAR.EQ.0) GO TO 1100	ABARR042
C		ABARR043
	NHI=IHR2	ABARR044
	IF (IHR1.GT.IHR2) NHI=24+IHR2	ABARR045
	HFS=NHI-IHR1+1	ABARR046
	DC 10 N=1, NABAR	ABARR047
	DC 10 I=1, NPLTS	ABARR048
	AEAR (I+5,N)=0.0	ABARR049
10	CCNTINUE	ABARR050
	T=5./9.*(TMBAR-32.0)+273.	ABARR051
	DC 20 J=1,7	ABARR052
	TVP (J)=EXP (ALPHA (J)-BETA (J)/T)	ABARR053
20	CCNTINUE	ABARR054
C		ABARR055
	IF (NWRK.EQ.0) GO TO 100	ABARR056
C	USING DEFAULT ACTIVITY FRACTIONS CALCULATE HC	ABARR057
C	EMISSIONS FROM ALL FUELS AND SPILLAGE.	ABARR058
C	ACCUMULATE IN ABAR AREAY	ABARR059
C		ABARR060
	DC 50 N=1, NWRK	ABARR061

```

HC=0.
FFC=0.
DC 40 J=1,7
FLHOUR(J)=0.
DC 30 I=IHR1,NHI
II=I
IF (I.GT.24) II=I-24
30  FLHOUR(J)=FLHOUR(J)+FLHR(II,J)
    FLHOUR(J)=FLHOUR(J)/HRS
    FFC=FRC+FLHOUR(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
    HC=HC+HCWRK(J+2,N)*TVP(J)*FLMO(IMONTH,J)*FLDY(IDAY,J)
    *FLHOUR(J)*7./DAYS
40  CONTINUE
    FRC=FRC/4.*7./DAYS
C
    J=HCWRK(2,N)
    ABAR(7,J)=ABAR(7,J) + (HC + HCWRK(10,N) * FRC) *(1.E+6/3.6)
50  CCNTINUE
C
100 IF (NERT.EQ.0) GO TO 200
C   CALCULATE HC EMISSIONS FROM SPECIFIED FUEL AND
C   TANK TYPES.  ACCUMULATE IN ABAR ARRAY
C
    DC 110 N=1,NBRT
    J=HCBRT(3,N)
    EX=0.68
    IF (HCBRT(4,N).EQ.2.) EX=0.70
    HC=HCBRT(5,N)*(TVP(J)/(14.7-TVP(J)))*EX*(1.E+6/(3.6*24.*365.))
C
    J=HCBRT(2,N)
    ABAR(7,J)=ABAR(7,J)+HC
110 CCNTINUE
C
200 IF (NXEVP.EQ.0) GO TO 300
    ICLASS=110
    NIEMF=NELTS
    NELTS=1
    ICC1=3
    NSRCE=NABARS
    DC 210 N=1,NXEVP
    DC 210 I=1,3
    SCRCE(I,NABARS+N)=HCEVE(I,N)
210 CONTINUE
    ICPT=1
    IF (JFLAG.EQ.0) READ 2,IOPT
    2  FORNAT(I4)
    GO TO (220,230),IOPT
C
220 CALL METHA(NXEVP,SORCE,I1,I2,ICLASS)
    GO TO 240
230 CALL METHC(NXEVP,SORCE,I1,I2,ICLASS)
C
240 DC 250 N=1,NXEVP
C
    ACCUMULATE OTHER EVAPORATIVE HC EMISSIONS IN ABAR ARRAY
C
    J=HCEVE(2,N)
    ABAR(7,J)=ABAR(7,J)+SCRCE(3,NABARS+N)
250 CONTINUE
C
    NELTS=NTEMP
    NSRCE=0

```

```

ABARR062
ABARR063
ABARR064
ABARR065
ABARR066
ABARR067
ABARR068
ABARR069
ABARR070
ABARR071
ABARR072
ABARR073
ABARR074
ABARR075
ABARR076
ABARR077
ABARR078
ABARR079
ABARR080
ABARR081
ABARR082
ABARR083
ABARR084
ABARR085
ABARR086
ABARR087
ABARR088
ABARR089
ABARR090
ABARR091
ABARR092
ABARR093
ABARR094
ABARR095
ABARR096
ABARR097
ABARR098
ABARR099
ABARR100
ABARR101
ABARR102
ABARR103
ABARR104
ABARR105
ABARR106
ABARR107
ABARR108
ABARR109
ABARR110
ABARR111
ABARR112
ABARR113
ABARR114
ABARR115
ABARR116
ABARR117
ABARR118
ABARR119
ABARR120
ABARR121
ABARR122
ABARR123

```

LCC1=2	ABARR124
C	ABARR125
300 IF (NSHS.EQ.0) GO TO 400	ABARR126
ICLASS=111	ABARR127
CALL METHB(NSHS,SORCE,I1,I2,ICLASS)	ABARR128
C	ABARR129
400 IF (NCFVHS.EQ.0) GO TO 500	ABARR130
ICLASS=112	ABARR131
CALL METHC(NCFVHS,SOPCE,I1,I2,ICLASS)	ABARR132
C	ABARR133
500 IF (NMVHAF.EQ.0) GO TO 600	ABARR134
CALL METHE(NMVHAF,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)	ABARR135
C	ABARR136
600 IF (NCVHAR.EQ.0) GO TO 700	ABARR137
CALL METHE(NCVHAR,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)	ABARR138
C	ABARR139
C****EMISSIONS ARE NOW IN MICROGRAMS/SEC	ABARR140
C FILL ABAR ARRAY WITH AREA GEOMETRIES	ABARR141
C	ABARR142
700 DO 710 N=1,NABAR	ABARR143
DC 710 I=1,5	ABARR144
ABAR(I,N)=ABAFGM(I+2,N)	ABARR145
710 CONTINUE	ABARR146
C	ABARR147
C FILL ABAR ARRAY WITH THE NON-EVAP HC EMISSION DATA	ABARR148
C	ABARR149
I1=11	ABARR150
I2=100	ABARR151
NSRCF=0	ABARR152
LCC1=5	ABARR153
IF (NSHS.EQ.0) GO TO 800	ABARR154
CALL EMISAR(NSHS,ABAR,I1,I2)	ABARR155
C	ABARR156
800 IF (NORVHS.EQ.0) GO TO 900	ABARR157
CALL EMISAR(NCFVHS,ABAR,I1,I2)	ABARR158
C	ABARR159
900 IF (NMVHAF.EQ.0) GO TO 1000	ABARR160
CALL EMISAR(NMVHAR,ABAR,I1,I2)	ABARR161
C	ABARR162
1000 IF (NCVHAR.EQ.0) GO TO 1100	ABARR163
CALL EMISAR(NCVHAR,ABAR,I1,I2)	ABARR164
C	ABARR165
1100 CONTINUE	ABARR166
RETURN	ABARR167
END	ABARR168



## SUBROUTINE ABLNAR

### Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft line sources.
2. To compute the emission rates due to military and civilian vehicle line and other airbase line activities.

### Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those other airbase line activities not using the default of a uniform distribution.

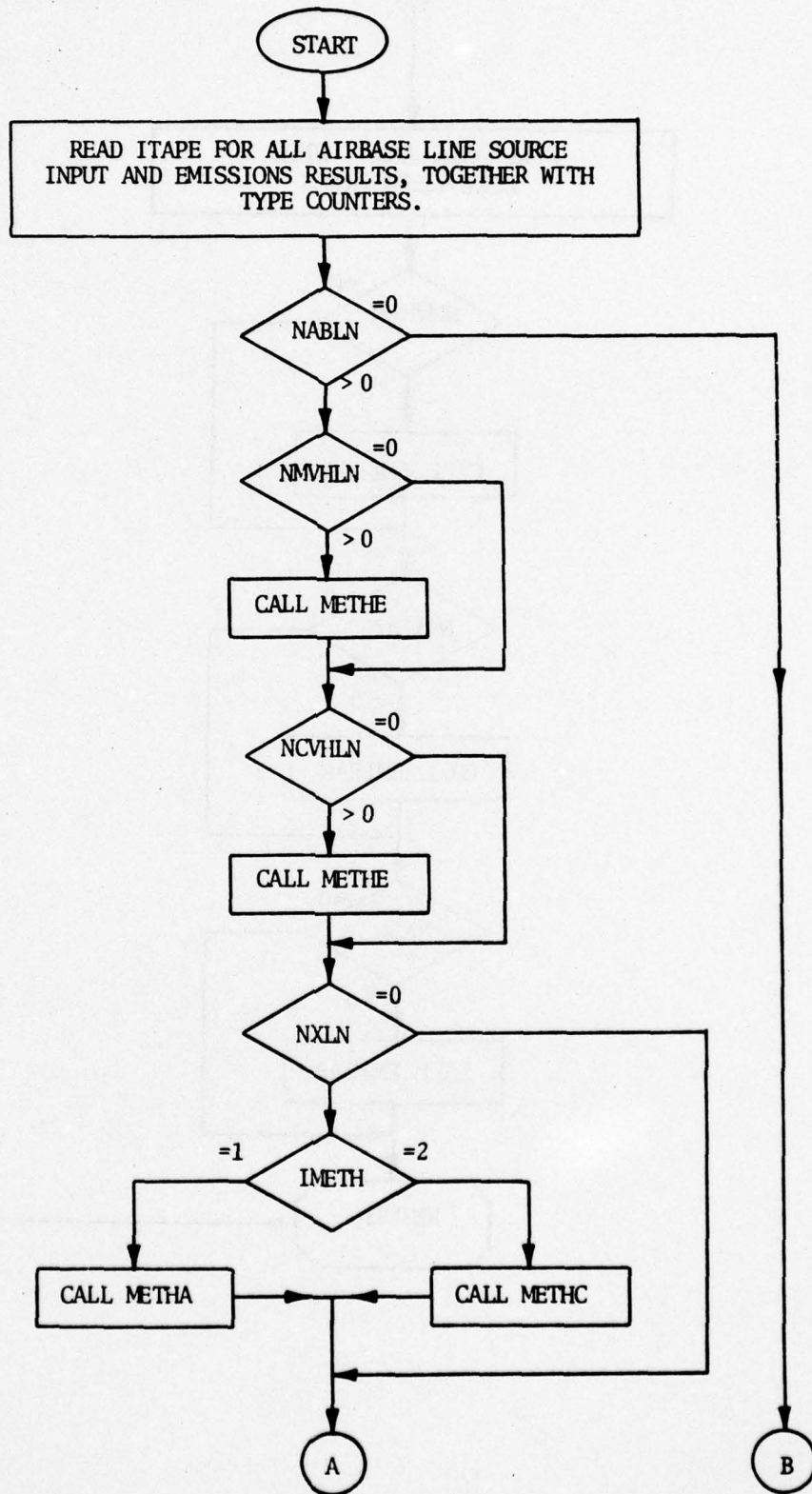
### Output:

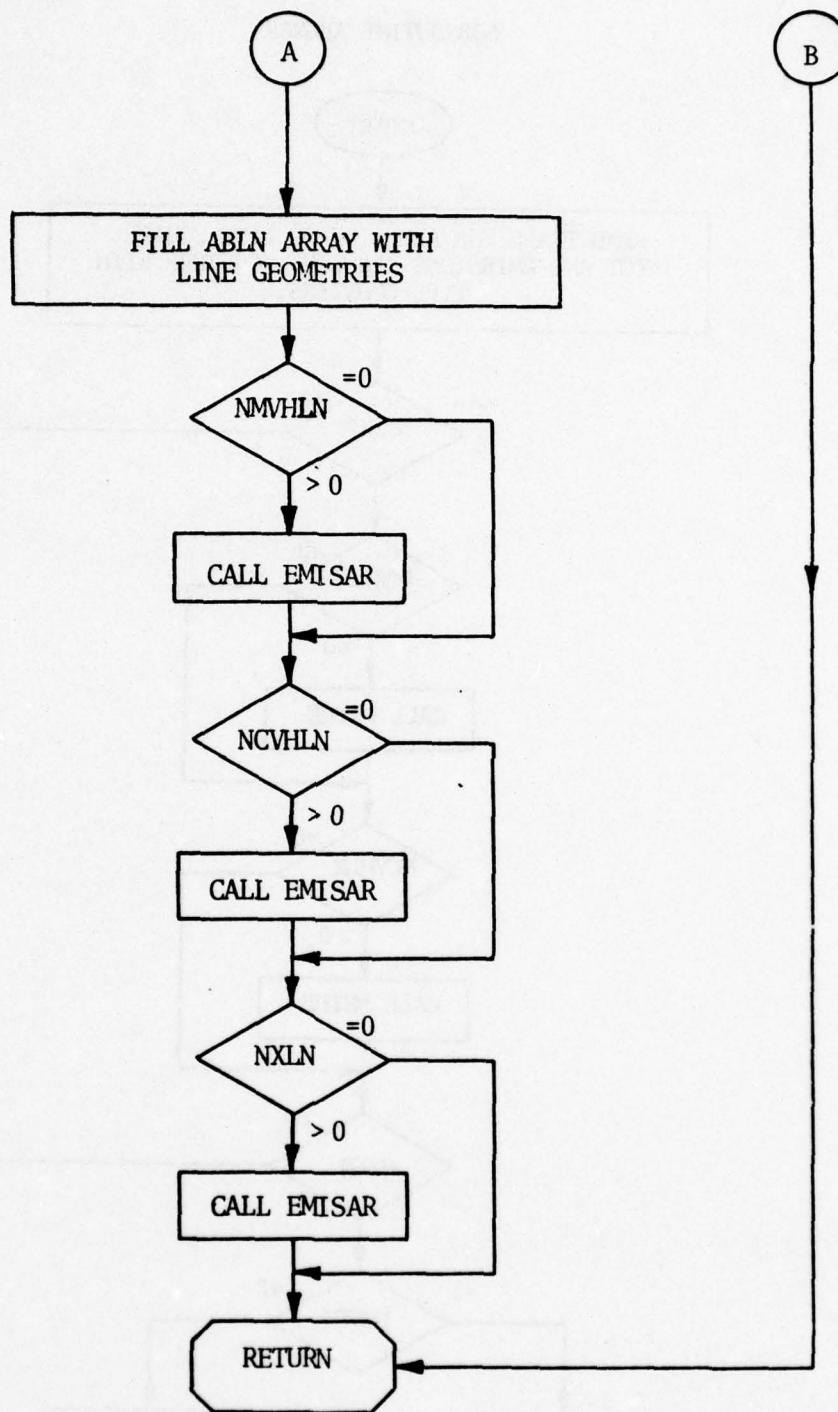
The array, ABLN, is filled with geometry and emission data for non-aircraft line sources.

### Subroutines Called:

METHA, METHC, METHE, EMISAR

# SUBROUTINE ABLNAR







C	SUBROUTINE ABLNAR	ABLNR000
C	THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL	ABLNR001
C	AIRBASE LINES	ABLNR002
C	NMVHLN = NO. OF MILITARY LINE ACTIVITIES	ABLNR003
C	NCVHLN = NO. OF CIVILIAN LINE ACTIVITIES	ABLNR004
C	NXLN = NO. OF OTHER AIR BASE LINE ACTIVITIES	ABLNR005
C		ABLNR006
	COMMON / DEFAULT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	ABLNR007
	COMMON / PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG	ABLNR008
	COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGH(10,200)	ABLNR009
	,LOC1,LOC2,NGEOM,IPT	ABLNR010
	COMMON /DSTRET/ ACHO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),	ABLNR011
	VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),	ABLNR012
	CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1	ABLNR013
	COMMON /SPCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,	ABLNR014
	NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),	ABLNR015
	ABPT(16,150),ABAR(11,100),ABLN(14,100)	ABLNR016
	LOC1=2	ABLNR017
	LOC2=2	ABLNR018
	I1=17	ABLNR019
	I2=300	ABLNR020
	NGEOM=0	ABLNR021
	IPT=0	ABLNR022
	NSRCE=0	ABLNR023
	PEAD(ITAPE) NABLN,NTOT,NMVHLN,NCVHLN,NXLN,NABLNS,	ABLNR024
	((SORGH(I,N),I=1,10),N=1,NABLN),	ABLNR025
	((SORCE(I,N),I=1,NTCT),N=1,NABLNS)	ABLNR026
C	IF (NABLN.EQ.0) GO TO 600	ABLNR027
	IF (NMVHLN.EQ.0) GO TO 100	ABLNR028
C	CALL METHE(NMVHLN,SORCE,VHMLMO,VHMLDY,VHMLHR,I1,I2)	ABLNR029
C	100 IF (NCVHLN.EQ.0) GO TO 200	ABLNR030
C	CALL METHE(NCVHLN,SORCE,CVABMO,CVABDY,CVABHR,I1,I2)	ABLNR031
C	200 IF (NXLN.EQ.0) GO TO 300	ABLNR032
	ICLASS=117	ABLNR033
C	IMETH=1	ABLNR034
	IF (JFLAG.EQ.0) READ 1,IMETH	ABLNR035
	1 FORMAT(I4)	ABLNR036
	GO TO (210,220),IMETH	ABLNR037
C	210 CALL METHA(NXLN,SORCE,I1,I2,ICLASS)	ABLNR038
	GO TO 300	ABLNR039
C	220 CALL METHC(NXLN,SORCE,I1,I2,ICLASS)	ABLNR040
C	C*****EMISSIONS ARE NOW IN MICROGRAMS/SEC	ABLNR041
C	FILL ABLN ARRAY WITH LINE GEOMETRIES	ABLNR042
C	300 DO 320 N=1,NABLN	ABLNR043
	DO 310 I=1,8	ABLNR044
	ABLN(I,N)=SORGH(I+2,N)	ABLNR045
	310 CONTINUE	ABLNR046
	DO 320 I=1,NPLTS	ABLNR047
	ABLN(I+8,N)=0.0	ABLNR048
	320 CONTINUE	ABLNR049
C		ABLNR050
		ABLNR051
		ABLNR052
		ABLNR053
		ABLNR054
		ABLNR055
		ABLNR056
		ABLNR057
		ABLNR058
		ABLNR059
		ABLNR060
		ABLNR061

C     FILL ABLN ARRAY WITH LINE EMISSION DATA

C

NSRCE=0

LOC1=8

I1=14

I2=100

IF (NMVHLN.EQ.0) GO TO 400

CALL EMISAR(NMVHLN,ABLN,I1,I2)

C

400 IF (NCVHLN.EQ.0) GO TO 500

CALL EMISAR(NCVHLN,ABLN,I1,I2)

C

500 IF (NXLN.EQ.0) GO TO 600

CALL EMISAR(NXLN,ABLN,I1,I2)

C

600 CCNTINUE

RETURN

END

ABLNRO62

ABLNRO63

ABLNRO64

ABLNRO65

ABLNRO66

ABLNRO67

ABLNRO68

ABLNRO69

ABLNRO70

ABLNRO71

ABLNRO72

ABLNRO73

ABLNRO74

ABLNRO75

ABLNRO76

ABLNRO77

ABLNRO78

ABLNRO79

## SUBROUTINE ABPTAR

### Purpose:

1. To read from the master source tape all data needed to define airbase non-aircraft point sources.
2. To compute the emission rates due to training fires, test cells, run-up stands, power plants, incinerators, storage tanks and other airbase point source activities.

### Input:

None

### Output:

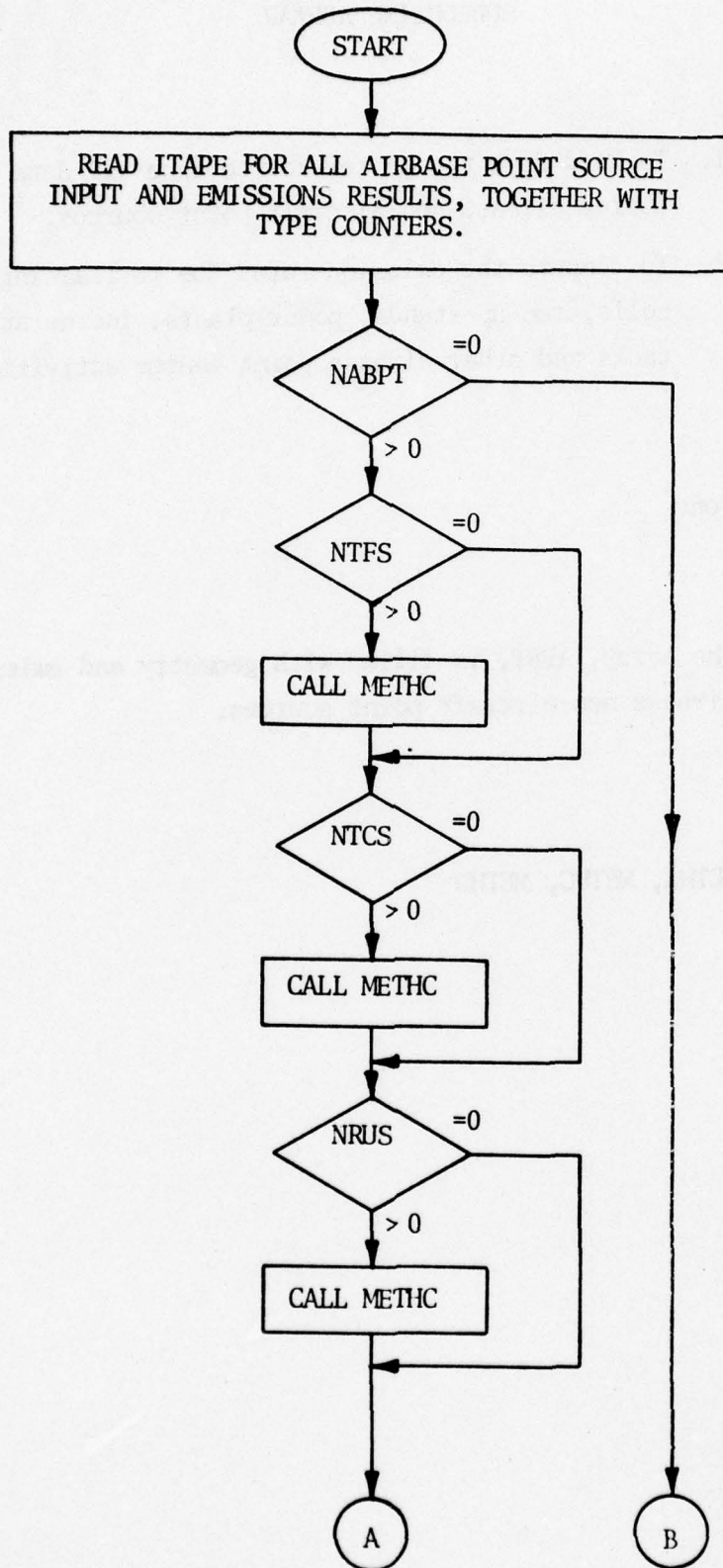
The array, APBT, is filled with geometry and emission data for airbase non-aircraft point sources.

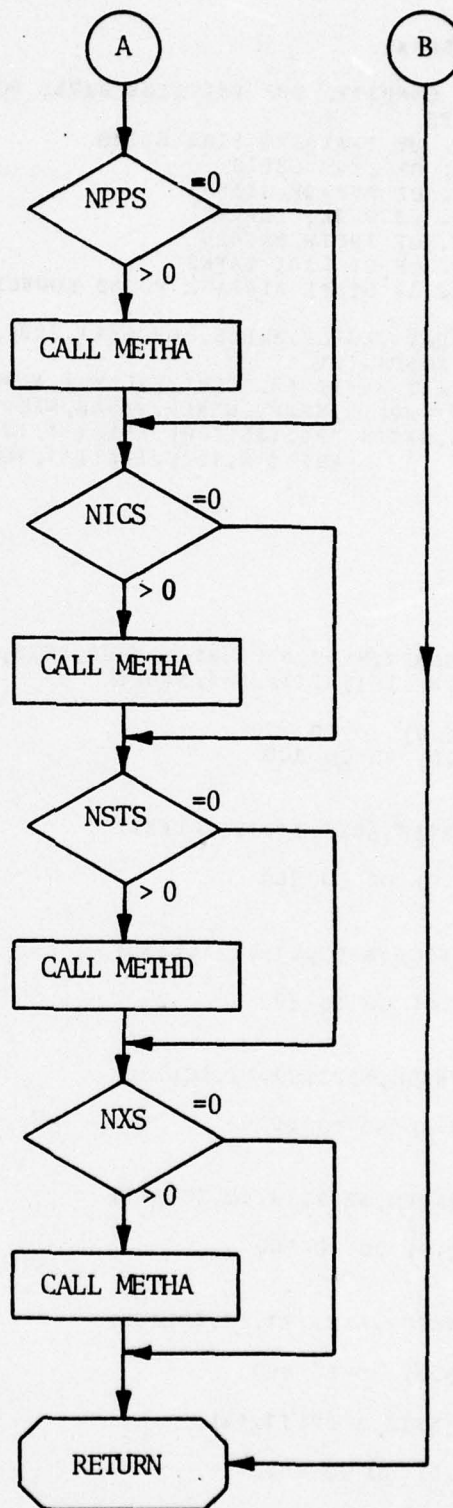
### Subroutines Called:

METHA, METHC, METHD



SUBROUTINE ABPTAR





C	SUBRCUTINE ABPTAR	ABPTR000
C	THIS RCUTINE COMPUTES THE EMISSION RATES FOR ALL	ABPTR001
C	AIRBASE POINTS	ABPTR002
C	NTFS = NO. OF TRAINING FIRE SITES	ABPTR003
C	NICS = NO. OF TEST CELLS	ABPTR004
C	NRUS = NO. OF RUN-UP STANDS	ABPTR005
C	NPFS = NO. OF POWER PLANTS	ABPTR006
C	NICS = NO. OF INCINERATORS	ABPTR007
C	NSIS = NO. OF STORAGE TANKS	ABPTR008
C	NXS = NO. OF OTHER AIRBASE POINT SOURCES	ABPTR009
C		ABPTR010
C	COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)	ABPTR011
C	LOC1,LOC2,NGEOM,IPT	ABPTR012
C	CCHMCN / DEFAULT / ITAFB,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	ABPTR013
C	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABIN,	ABPTR014
C	NACPT,NACAR,NACLN,ENET(16,100),ENAR(11,100),ENLN(14,20),	ABPTR015
C	ABFT(16,150),ABAR(11,100),ABLN(14,100)	ABPTR016
C	LCC1=10	ABPTR017
C	LCC2=11	ABPTR018
C	NGEOM=9	ABPTR019
C	IFT=1	ABPTR020
C	NSRCE=0	ABPTR021
C	I1=16	ABPTR022
C	I2=200	ABPTR023
C	READ(ITAPE) NABPT,NTOT,NTFS,NTCS,NRUS,NPPS,NICS,NSIS,NXS,	ABPTR024
C	((SCFCE(I,N),I=1,NTCT),N=1,NABPT)	ABPTR025
C	IF (NABFT.EQ.0) GO TO 700	ABPTR026
C	IF (NTFS.EQ.0) GO TO 100	ABPTR027
C	ICLASS=101	ABPTR028
C	CALL METHC(NTFS,ABPT,I1,I2,ICLASS)	ABPTR029
C	100 IF (NICS.EQ.0) GO TO 200	ABPTR030
C	ICLASS=102	ABPTR031
C	CALL METHC(NICS,ABPT,I1,I2,ICLASS)	ABPTR032
C	200 IF (NRUS.EQ.0) GO TO 300	ABPTR033
C	ICLASS=103	ABPTR034
C	CALL METHC(NRUS,ABPT,I1,I2,ICLASS)	ABPTR035
C	300 IF (NPFS.EQ.0) GO TO 400	ABPTR036
C	ICLASS=104	ABPTR037
C	CALL METHA(NPPS,ABPT,I1,I2,ICLASS)	ABPTR038
C	400 IF (NICS.EQ.0) GO TO 500	ABPTR039
C	ICLASS=105	ABPTR040
C	CALL METHA(NICS,ABPT,I1,I2,ICLASS)	ABPTR041
C	500 IF (NSIS.EQ.0) GO TO 600	ABPTR042
C	CALL METHD(NSIS,ABPT,I1,I2)	ABPTR043
C	600 IF (NXS.EQ.0) GO TO 700	ABPTR044
C	ICLASS=107	ABPTR045
C	CALL METHA(NXS,ABPT,I1,I2,ICLASS)	ABPTR046
		ABPTR047
		ABPTR048
		ABPTR049
		ABPTR050
		ABPTR051
		ABPTR052
		ABPTR053
		ABPTR054
		ABPTR055
		ABPTR056
		ABPTR057
		ABPTR058
		ABPTR059
		ABPTR060
		ABPTR061



C  
700 RETURN  
END

ABPTR062  
ABPTR063  
ABPTR064

## SUBROUTINE ACSRCE

### Purpose:

To set up the aircraft source arrays to be used by the dispersion routines for calculating ground level concentrations.

### Input:

Basic aircraft data, airbase activity data, points in arrival-departure paths and in training flight paths, meteorological conditions, time period of calculation.

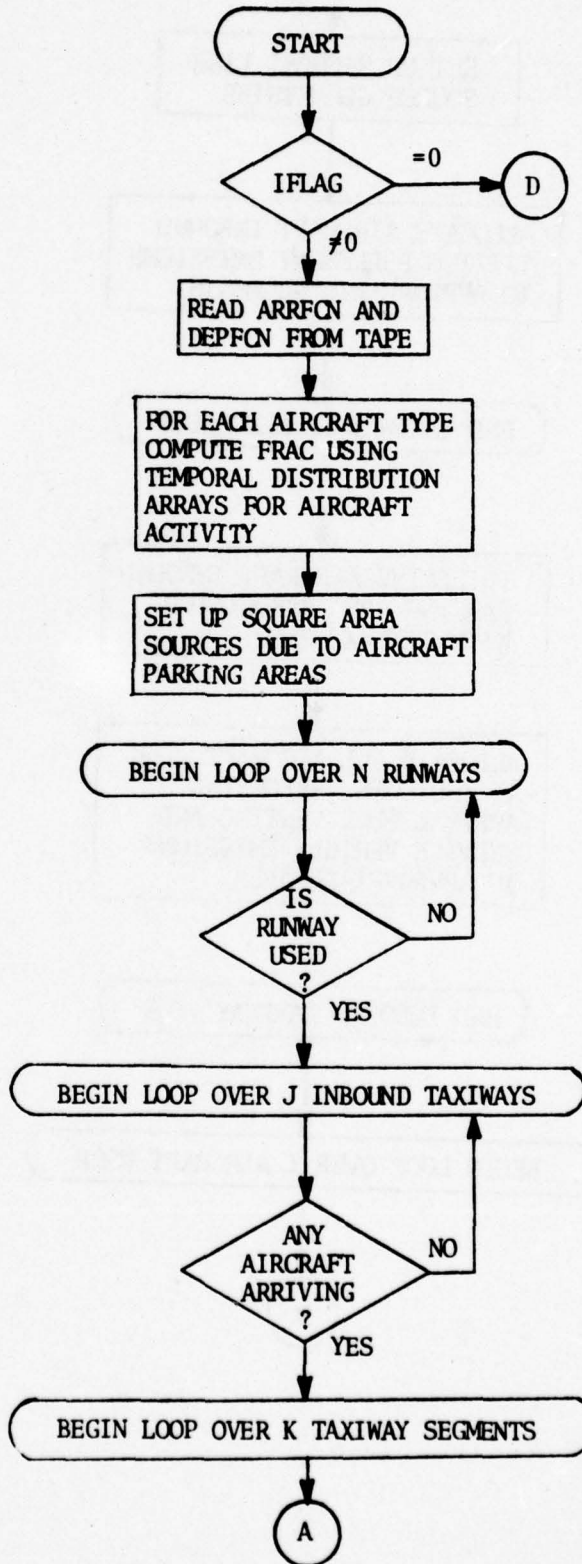
### Output:

The arrays ACPT, ACLN and ACAR to contain all source information necessary to calculate dispersion and pollutant concentrations.

### Subroutine Called:

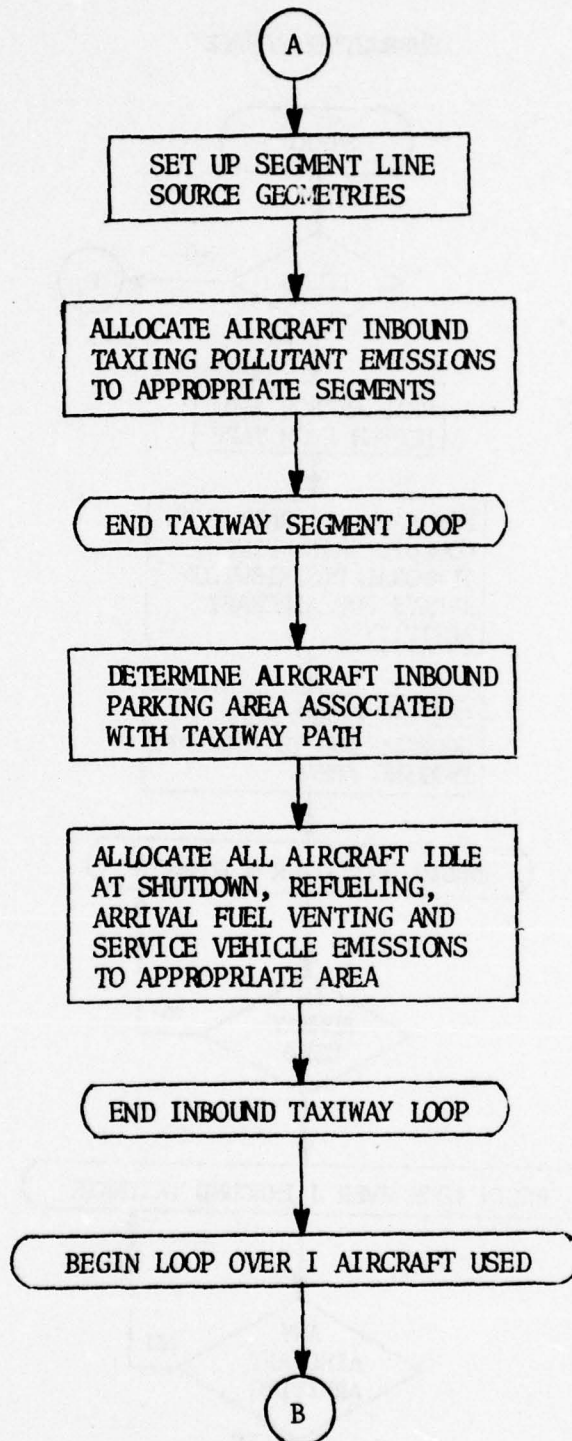
DEPART

SUBROUTINE ACSRCE

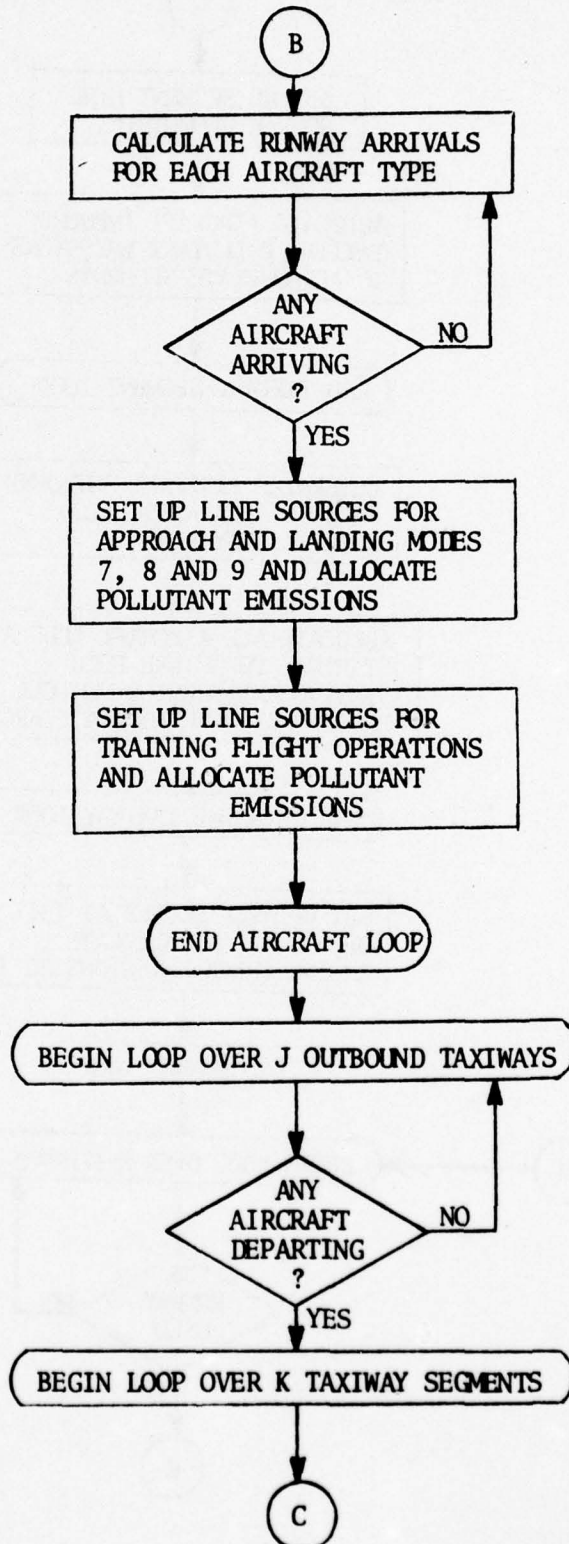




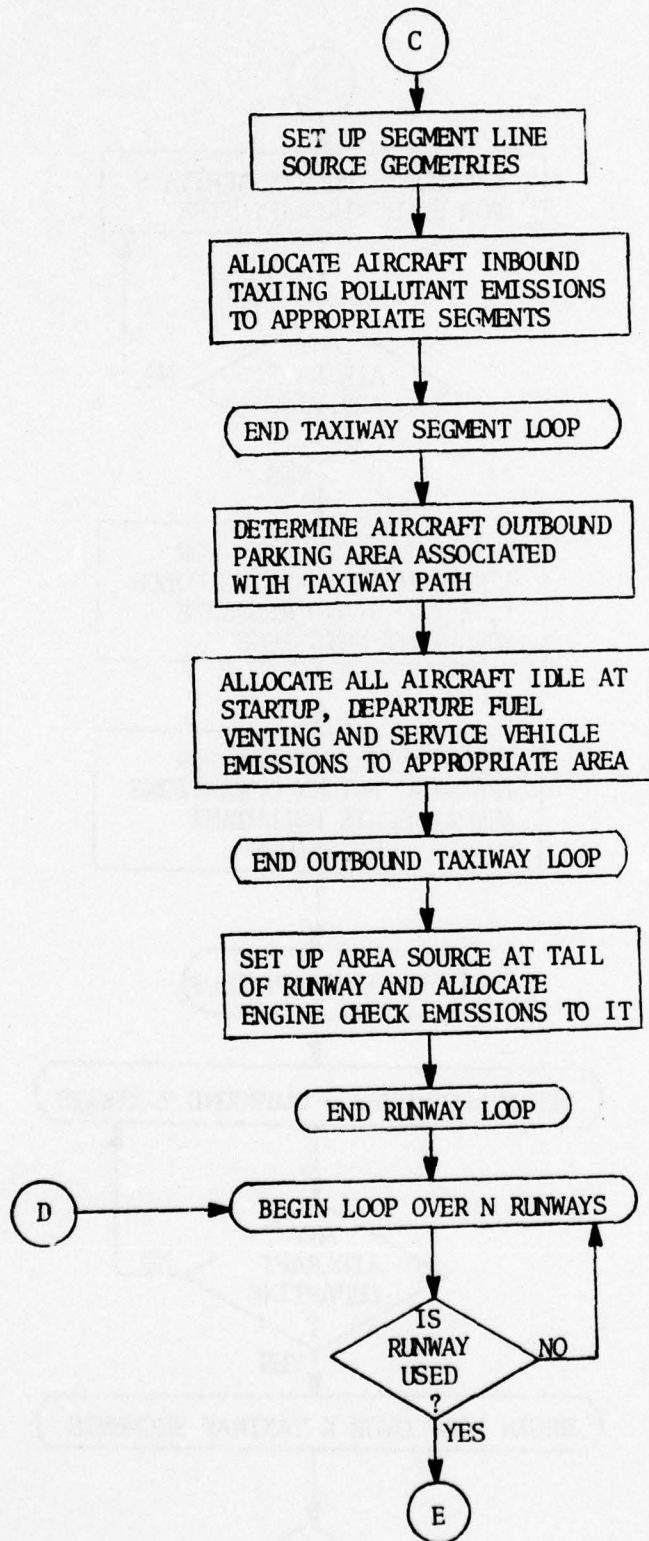
SUBROUTINE ACSRCE (Cont'd.)



SUBROUTINE ACSRCE (Cont'd.)

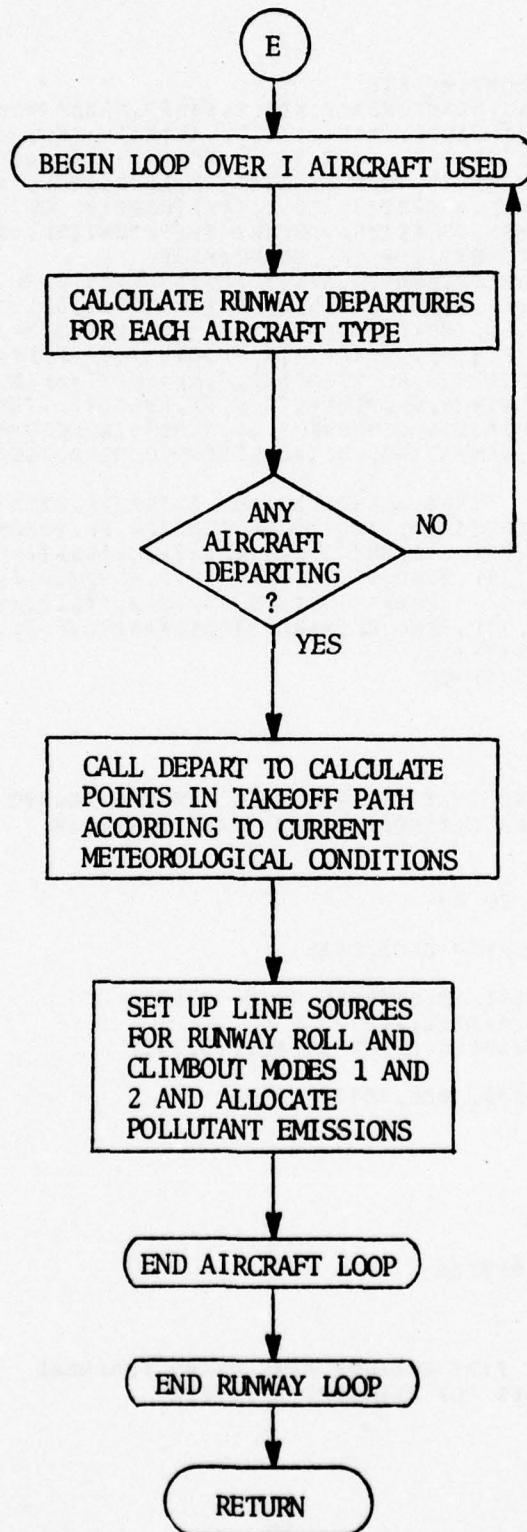


SUBROUTINE ACSRCE (Cont'd.)





SUBROUTINE ACSRCE (Cont'd.)



C	SUBROUTINE ACSRCE	ACSR000
C	THIS ROUTINE SETS UP THE AIRCRAFT SOURCE ARRAYS	ACSR001
C	AND ALLOCATES THE POLLUTANT EMISSIONS TO THE	ACSR002
C	APPROPRIATE AREA OR LINE	ACSR003
C		ACSR004
	REAL LNDSPD	ACSR005
	INTFGER ENGNO	ACSR006
	COMMON /RECPT/ MRECPT,MAXFIL	ACSR007
	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,	ACSR008
	. NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),	ACSR009
	. ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)	ACSR010
	COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),	ACSR011
	. LNDSPD(8),APSPD1(8),APSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),	ACSR012
	. COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHK1(8),SHTDNT(8),DSCNT2(8),	ACSR013
	. APPHT,APPHT2(8),CLMBPT,TOWT(8),ENGNO(8,2)	ACSR014
	COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNARR(8),	ACSR015
	. ANNDEP(8),ANNTOGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),	ACSR016
	. DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),	ACSR017
	. ACPIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),	ACSR018
	. IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),	ACSR019
	. IOBSEG(16,8,6),IOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),	ACSR020
	. PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)	ACSR021
	COMMON / NET / WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMF,	ACSR022
	1 TEMK	ACSR023
	COMMON / DEFALT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	ACSR024
	COMMON /DSTRET/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),	ACSR025
	. VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),	ACSR026
	. CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1	ACSR027
	COMMON /PERIOD/ IMO ,NODAYS,IDY ,IHR1,IHR2,IFLAG,JFLAG	ACSR028
	DIMENSION IACAR(2,18),FRAC(8),PARFCT(18),APARSQ(6,3),NQ(25)	ACSR029
	XP(XO,YC,W)=YC*SIN(W)+XO	ACSR030
	YP(YO,YC,W)=YC*COS(W)+YO	ACSR031
	DAYS=NODAYS	ACSR032
	NT=NPLTS+5	ACSR033
	IWIND=29+IWD	ACSR034
C		ACSR035
C	AN IFLAG OF 0 MEANS THAT ALL AIRCRAFT SOURCES EXCEPT	ACSR036
C	FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2 REMAIN	ACSR037
C	UNCHANGED	ACSR038
C		ACSR039
	IF(IFLAG.EQ.0) GO TO 69	ACSR040
C		ACSR041
C	READ ARRFEN AND DEPFCN FROM TAPE	ACSR042
C		ACSR043
	IF (IWD.GE.1.AND.IWD.LE.MAXFIL) GO TO 1000	ACSR044
	PRINT 9000,MRECPT,MAXFIL,IWD	ACSR045
9000	FORMAT(29HCFE REQUEST ERROR IN ACSRCE,315)	ACSR046
	GO TO 1040	ACSR047
1000	IF (MRECPT-IWD) 1010,1030,1020	ACSR048
1010	READ (30)	ACSR049
	MRECPT=MRECPT+1	ACSR050
	GO TO 1000	ACSR051
1020	REWIND 30	ACSR052
	MRECPT=1	ACSR053
	GO TO 1000	ACSR054
1030	READ (30) ARRFEN,DEPFCN	ACSR055
	MRECPT=MRECPT+1	ACSR056
1040	CONTINUE	ACSR057
C		ACSR058
C	FOR EACH AIRCRAFT TYPE COMPUTE FRAC USING TEMPORAL	ACSR059
C	DISTRIBUTION ARRAYS FOR AIRCRAFT ACTIVITY	ACSR060
		ACSR061

C	NHI=IHR2	ACSRC062
	IF (IHR1.GT.IHR2) NHI=24+IHR2	ACSRC063
	HRS=NHI-IHR1+1	ACSPC064
	DC 5 I=1, NACTYP	ACSRC065
	HRFRC=0.	ACSRC066
	DC 4 JJ=IHR1, NHI	ACSRC067
	J=JJ	ACSPC068
	IF (JJ.GT.24) J=JJ-24	ACSRC069
4	HRFRC=HFFRC+ACHP (J, I)	ACSRC070
	HRFRC=HRFRC/HFS	ACSRC071
	FRAC (I)=ACMO (IMO, I) *ACDY (IDY, I) *HRFRC*7.0/DAYS*(1.E+6/3.6)	ACSPC072
5	CONTINUE	ACSRC073
8	NACPI=0	ACSRC074
	NB=C	ACSRC075
	NC=0	ACSRC076
	NZ=0	ACSRC077
C		ACSRC078
C	SET UP SQUARE AREA SOURCES DUE TO AIRCRAFT PARKING AREAS	ACSRC079
C		ACSRC080
	DO 1 L=1, NPKAR	ACSRC081
	NSQ=NPASQ (L)	ACSRC082
	SPARSQ=0.0	ACSRC083
	DO 2 J=1, NSQ	ACSRC084
	NB=NB+1	ACSRC085
	ACAP (1, NB)=PAREA (L, J, 1)	ACSPC086
	ACAP (2, NB)=PAREA (L, J, 2)	ACSRC087
	ACAR (3, NB)=ACLNDZ/2.	ACSRC088
	ACAR (4, NB)=PAREA (L, J, 3) *1000.	ACSRC089
	AEAFSQ (L, J)=ACAR (4, NB) ** 2	ACSRC090
	SPARSQ = SPARSQ + APARSQ (L, J)	ACSRC091
	ACAR (5, NB)=ACLNDZ	ACSRC092
	IACAP (1, NB)=IDPRKA (L)	ACSRC093
2	IACAP (2, NB)=NSQ	ACSRC094
	DO 91 J=1, NSQ	ACSRC095
	NZ=NZ+1	ACSRC096
91	FAFFCT (NZ) = APARSQ (L, J) / SPARSQ	ACSRC097
1	CONTINUE	ACSRC098
C		ACSPC099
	DO 93 I=1, NLSEGS	ACSRC100
93	NQ (I)=0	ACSPC101
	NEKSRC=NB	ACSRC102
	DO 3 L=1, NPKSRC	ACSRC103
	DC 3 K=6, NT	ACSPC104
	HRACAR (K-5, L)=0.0	ACSRC105
3	ACAR (K, L)=0.0	ACSRC106
	TVP= EXP (ALPHA (2) -BETA (2) /TEMK)	ACSRC107
C		ACSRC108
C	BEGIN LOOP OVER N RUNWAYS	ACSPC109
C		ACSRC110
	DC 10 N=1, NRNWYS	ACSRC111
C		ACSPC112
C	IS RUNWAY USED WITH THIS WIND DIRECTION?	ACSRC113
C		ACSRC114
	IF (1USWD (IWD, N).EQ.0) GO TO 10	ACSRC115
	THETA=RNWY (7, N)	ACSRC116
	XO=0.25*DISRNW (N)*SIN (THETA)+RNWY (2, N)	ACSRC117
	YO=0.25*DISRNW (N)*COS (THETA)+RNWY (3, N)	ACSRC118
	NTT=NIBTT (N)	ACSRC119
	IF (NTT.EQ.0) GO TO 50	ACSRC120
C		ACSRC121
C	BEGIN LOOP OVER J INBOUND TAXIWAYS	ACSRC122
		ACSRC123



C	DC 11 J=1,NTT	ACSRC124
C		ACSRC125
C	ANY AIRCRAFT ARRIVING ON THIS RUNWAY?	ACSRC126
C		ACSRC127
	DC 7 I=1,NACTYP	ACSRC128
	IF(TTARFR(J,I,N)*ARRFCN(23,I,N).GT.0.0) GO TO 701	ACSRC129
7	CCONTINUE	ACSRC130
	GC TO 11	ACSRC131
701	NSGLNS = NIBSEG (J,N)	ACSRC132
C		ACSRC133
C	BEGIN LOOP OVER K TAXIWAY SEGMENTS	ACSRC134
C		ACSRC135
	DC 12 K=1,NSGLNS	ACSRC136
C		ACSRC137
C	SET UP SEGMENT LINE SOURCE GEOMETRIES	ACSRC138
C		ACSRC139
	JJ = IIBSEG(K,J,N)	ACSRC140
	IF(NQ(JJ).NE.0) GO TO 130	ACSRC141
	NC=NC+1	ACSRC142
	NQ(JJ)=NC	ACSRC143
	DO 121 L=1, 12	ACSRC144
121	ACLN(L,NC)=ACLNSG(L,JJ)	ACSRC145
	ACLN(9,NC)=1.0	ACSRC146
	ACLN(10,NC)=1.0	ACSRC147
C		ACSRC148
C	ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS	ACSRC149
C	TO APPROPRIATE SEGMENTS	ACSRC150
C		ACSRC151
	DO 13 L=1,NPLTS	ACSRC152
	LL=L+12	ACSRC153
13	ACLN(LL,NC)=0.0	ACSRC154
130	ND=NQ(JJ)	ACSRC155
	DC 14 I=1,NACTYP	ACSRC156
	AA=ENGNO(I, 1)	ACSRC157
	IF(IEGFLG.GT.0) AA=ENGNO(I, 2)	ACSRC158
	ARR=TTARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)	ACSRC159
	IF(ARR.LE.0.0)GO TO 14	ACSRC160
	TIME=ACLN(11,ND)/(TXISFD(I)*ACLNSG(9,JJ))	ACSRC161
	FRC=AA*ARR*TIME*FRAC(I)	ACSRC162
	DC 15 L=1,NPLTS	ACSRC163
	KK=L+12	ACSRC164
15	ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I, 2,L)	ACSRC165
14	CCONTINUE	ACSRC166
12	CONTINUE	ACSRC167
C		ACSRC168
C	END TAXIWAY SEGMENT LOOP	ACSRC169
C		ACSRC170
C		ACSRC171
C	DETERMINE AIRCRAFT INBOUND PARKING AREA	ACSRC172
C	ASSOCIATED WITH TAXIWAY PATH	ACSRC173
C		ACSRC174
	DO 16 I=1,NPKSRC	ACSRC175
	II=I	ACSRC176
	IDPK=IACAR(1,I)	ACSRC177
	IF(IDPK.EQ.IDIBPA(J,N))GO TO 17	ACSRC178
16	CONTINUE	ACSRC179
	PRINT 18, IDIBPA(J,N),J,N	ACSRC180
18	FORMAT ('OINBCUND PARKING AREA 'I3,'OF TAXIWAY='I3,'; RUNWAY='I3,'	ACSRC181
	1 IS NOT CONSISTANT WITH PARKING AREA ID NUMBERS')	ACSRC182
	STOF	ACSRC183
17	CONTINUE	ACSRC184
		ACSRC185

C		ACSRC186
C	ALLOCATE ALL AIRCRAFT IDLE AT SHUTDOWN, REFUELING,	ACSRC187
C	ARRIVAL FUEL VENTING AND SERVICE VEHICLE EMISSIONS	ACSRC188
C	TO APPROPRIATE AREA	ACSRC189
C		ACSRC190
	NSQ=IACAR(2,II)	ACSRC191
	DC 19 I=1,NACTYP	ACSRC192
	ARR=TIARFR(J,I,N)*ARRFCN(23,I,N)*ANNARR(I)	ACSRC193
	IF(ARR.LE.0.0) GO TO 19	ACSRC194
	AA=ENGNO(I,1)	ACSRC195
	IF(IEGFLG.GT.0) AA=ENGNO(I,2)	ACSRC196
	TIME=SHIDNT(I)/60.	ACSRC197
	FRC=AA*ARR*TIME*FRAC(I)	ACSRC198
	TVP=EXP(ALPHA(JES1(I)) - BETA(JES1(I)) / TEMK)	ACSRC199
	DC 20 I=1,NSQ	ACSRC200
	JJ=II+L-1	ACSRC201
	DO 21 K=1,NPLTS	ACSRC202
	KK=K+5	ACSRC203
	ACAR(KK,JJ)=ACAR(KK,JJ)+FRC*ACEMFC(I,1,K) * PARFCT(JJ)	ACSRC204
	ACAR(KK,JJ)=ACAR(KK,JJ) + (ARSVEM(K,I,1) + ARSVEM(K,I,2) +	ACSRC205
	. ARSVEM(K,I,3) + ARSVEM(K,I,4)+ARSVEM(K,I,5)) * ARR * FRAC(I)	ACSRC206
	. * PARFCT(JJ)	ACSRC207
	IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+(0.3*TVP*ACFUEL(I)*0.5	ACSRC208
	1/1000. + ACSPIL(I) + ARFLVT(I)) * ARR * FLDENS(JES1(I)) * FRAC(I)	ACSRC209
	. * PARFCT(JJ)	ACSRC210
21	CONTINUE	ACSRC211
20	CONTINUE	ACSRC212
19	CONTINUE	ACSRC213
11	CONTINUE	ACSRC214
C		ACSRC215
C	END INBOUND TAXIWAY LOOP	ACSRC216
C		ACSRC217
C		ACSRC218
C	BEGIN LOOP OVER I AIRCRAFT USED	ACSRC219
C		ACSRC220
	DC 30 I=1,NACTYP	ACSRC221
C		ACSRC222
C	CALCULATE RUNWAY ARRIVALS FOR EACH AIRCRAFT TYPE	ACSRC223
C		ACSRC224
	ARR=ARRFCN(23,I,N)*ANNARR(I)	ACSRC225
C		ACSRC226
C	ANY AIRCRAFT ARRIVING?	ACSRC227
C		ACSRC228
	IF(ARR.LE.0.0) GO TO 30	ACSRC229
C		ACSRC230
C	SET UP LINE SOURCES FOR APPROACH AND LANDING MODES 7, 8 AND 9	ACSRC231
C	AND ALLOCATE POLLUTANT EMISSIONS	ACSRC232
C		ACSRC233
	AA=ENGNC(I,1)	ACSRC234
	DC 31 J=1,3	ACSRC235
	DO 32 K=1,3	ACSRC236
	KK=K+NC	ACSRC237
	JK=6*K-6+J	ACSRC238
	ACLN(J,KK)=ARRFCN(JK,I,N)	ACSRC239
	ACLN(J+5,KK)=ARRFCN(JK+6,I,N)	ACSRC240
32	CONTINUE	ACSRC241
	JJ=NC+J	ACSRC242
	JK=6*J-2	ACSRC243
	ACLN(4,JJ)=ARRFCN(24,I,N)	ACSRC244
	ACLN(5,JJ)=DEPFCN(24,I,N)	ACSRC245
	ACLN(09,JJ)=ARRFCN(JK,I,N)	ACSRC246
	ACLN(10,JJ)=ARRFCN(JK+6,I,N)	ACSRC247

	ACLN(11,JJ)=ARRFCN(JK+1,I,N)	ACSRC248
	ACLN(12,JJ)=AFRFCN(JK+2,I,N)	ACSRC249
	JMODE=J+6	ACSRC250
	DO 33 K=1,NPLTS	ACSRC251
	KK=K+12	ACSRC252
33	ACLN(KK,JJ)=AA*ACEMFC(I,JMODE,K)*ARR*ARRFCN(JK+2,I,N)*FRAC(I)	ACSRC253
31	CCONTINUE	ACSRC254
	CONTINUE	ACSRC255
	NC=NC+3	ACSRC256
C		ACSRC257
C	SET UP LINE SOURCES FOR TRAINING FLIGHT OPERATIONS	ACSRC258
C	AND ALLOCATE POLLUTANT EMISSIONS	ACSRC259
C		ACSRC260
	IF(ANNTGO(I).LE.0.0) GO TO 30	ACSRC261
	NC=NC+1	ACSRC262
	ACLN(1,NC)=XP(XO,TGO(1,1,I),THETA)	ACSRC263
	ACLN(2,NC)=YP(YO,TGO(1,1,I),THETA)	ACSRC264
	ACLN(6,NC)=XP(XO,TGO(1,2,I),THETA)	ACSRC265
	ACLN(7,NC)=YP(YO,TGO(1,2,I),THETA)	ACSRC266
	ACLN(1,NC+1)=ACLN(6,NC)	ACSRC267
	ACLN(2,NC+1)=ACLN(7,NC)	ACSRC268
	ACLN(6,NC+1)=XO	ACSRC269
	ACLN(7,NC+1)=YO	ACSRC270
	ACLN(1,NC+2)=XO	ACSRC271
	ACLN(2,NC+2)=YO	ACSRC272
	ACLN(6,NC+2)=XP(XO,0.3048,THETA)	ACSRC273
	ACLN(7,NC+2)=YP(YO,0.3048,THETA)	ACSRC274
	ACLN(1,NC+3)=ACLN(6,NC+2)	ACSRC275
	ACLN(2,NC+3)=ACLN(7,NC+2)	ACSRC276
	ACLN(6,NC+3)=XP(XO,TGO(1,3,I),THETA)	ACSRC277
	ACLN(7,NC+3)=YP(YO,TGO(1,3,I),THETA)	ACSRC278
	ACLN(1,NC+4)=ACLN(6,NC+3)	ACSRC279
	ACLN(2,NC+4)=ACLN(7,NC+3)	ACSRC280
	ACLN(6,NC+4)=XP(XO,TGO(1,4,I),THETA)	ACSRC281
	ACLN(7,NC+4)=YP(YO,TGO(1,4,I),THETA)	ACSRC282
	ACLN(3,NC)=APFHT*1000.	ACSRC283
	ACLN(8,NC)=APPHT2(I)*1000.	ACSRC284
	ACLN(3,NC+1)=APPHT2(I)*1000.	ACSRC285
	ACLN(8,NC+1)=ACLNDZ/2.	ACSRC286
	ACLN(3,NC+2)=ACLNDZ/2.	ACSRC287
	ACLN(8,NC+2)=ACLNDZ/2.	ACSRC288
	ACLN(3,NC+3)=ACLNDZ/2.	ACSRC289
	ACLN(8,NC+3)=COHT1(I)*1000.	ACSRC290
	ACLN(3,NC+4)=COHT1(I)*1000.	ACSRC291
	ACLN(8,NC+4)=CLMBHT*1000.	ACSRC292
	ACLN(09,NC)=APSPD1(I)	ACSRC293
	ACLN(10,NC)=APSPD2(I)	ACSRC294
	ACLN(11,NC)=TGO(2,1,I)	ACSRC295
	ACLN(12,NC)=TGO(3,1,I)	ACSRC296
	ACLN(09,NC+1)=APSPD2(I)	ACSRC297
	ACLN(10,NC+1)=LNDSPD(I)	ACSRC298
	ACLN(11,NC+1)=TGO(2,2,I)	ACSRC299
	ACLN(12,NC+1)=TGO(3,2,I)	ACSRC300
	ACLN(09,NC+2)=LNDSPD(I)*1.3	ACSRC301
	ACLN(10,NC+2)=TOSPD(I)*0.7	ACSRC302
	ACLN(11,NC+2)=0.3048	ACSRC303
	ACLN(12,NC+2)=2.0*0.3048/(1.3*LNDSPD(I)+0.7*TOSPD(I))	ACSRC304
	ACLN(09,NC+3)=TOSPD(I)	ACSRC305
	ACLN(10,NC+3)=COSPD1(I)	ACSRC306
	ACLN(11,NC+3)=TGO(2,3,I)	ACSRC307
	ACLN(12,NC+3)=TGO(3,3,I)	ACSRC308
	ACLN(09,NC+4)=COSPD1(I)	ACSRC309



	ACLN(10,NC+4)=COSPD2(I)	ACSRC310
	ACLN(11,NC+4)=TGO(2,4,I)	ACSRC311
	ACLN(12,NC+4)=TGO(3,4,I)	ACSRC312
	DO 45 J=1,5	ACSRC313
	JJ=NC+J-1	ACSRC314
	ACLN(4,JJ)=ARRFCN(24,I,N)	ACSRC315
	ACLN(5,JJ)=DEPFCN(24,I,N)	ACSRC316
	GO TO (34,35,41,36,37),J	ACSRC317
34	KD=7	ACSRC318
	GC IO 38	ACSRC319
35	KD=8	ACSRC320
	GO TC 38	ACSRC321
36	KD=5	ACSRC322
	GO TO 38	ACSRC323
37	KD=6	ACSRC324
38	DO 39 K=1,NPLTS	ACSRC325
	KK=K+12	ACSRC326
39	ACLN(KK,JJ)=ANNTGO(I)*ACEMFC(I,KD,K)*ARRFCN(23,I,N)*ACLN(12,JJ)*	ACSRC327
	1FRAC(I)*AA	ACSRC328
	GO TO 45	ACSRC329
41	DO 42 K=1,NPLTS	ACSRC330
	KK=K+12	ACSRC331
42	ACLN(KK,JJ)=AA*(0.3*ACEMFC(I,9,K)+0.7*ACEMFC(I,4,K))*	ACSRC332
	1 ANNTGC(I)*ARRFCN(23,I,N)*ACLN(12,JJ)*FRAC(I)	ACSRC333
45	CONTINUE	ACSRC334
	NC=NC+4	ACSRC335
30	CONTINUE	ACSRC336
C		ACSRC337
C	END AIRCRAFT LOOP	ACSRC338
C		ACSRC339
50	NTT=NOBTI(N)	ACSRC340
	IF(NTT.EQ.0) GO TO 10	ACSRC341
C		ACSRC342
C	BEGIN LOOP OVER J OUTBOUND TAXIWAYS	ACSRC343
C		ACSRC344
	DC 51 J=1,NTT	ACSRC345
C		ACSRC346
C	ANY AIRCRAFT DEPARTING ON THIS TAXIWAY?	ACSRC347
C		ACSRC348
	DC 6 I=1,NACTYP	ACSRC349
	IF(TIDPFR(J,I,N)*DEPFCN(23,I,N).GT.0.0) GO TO 601	ACSRC350
6	CONTINUE	ACSRC351
	GO TO 51	ACSRC352
601	NSGLNS=NOBSEG(J,N)	ACSRC353
C		ACSRC354
C	BEGIN LOOP OVER K TAXIWAY SEGMENTS	ACSRC355
C		ACSRC356
	DC 52 K=1,NSGLNS	ACSRC357
C		ACSRC358
C	SET UP SEGMENT LINE SOURCE GEOMETRIES	ACSRC359
C		ACSRC360
	JJ=IOBSEG(K,J,N)	ACSRC361
	IF(NQ(JJ).NE.0) GO TO 131	ACSRC362
	NC=NC+1	ACSRC363
	NQ(JJ)=NC	ACSRC364
	DC 122 L=1,12	ACSRC365
122	ACLN(L,NC)=ACLNSG(L,JJ)	ACSRC366
	ACLN(9,NC)=1.0	ACSRC367
	ACLN(10,NC)=1.0	ACSRC368
C		ACSRC369
C	ALLOCATE AIRCRAFT INBOUND TAXIING POLLUTANT EMISSIONS	ACSRC370
C	TO APPROPRIATE SEGMENTS	ACSRC371

C	DC 53 L=1,NPLTS	ACSRC372
	LL=L+12	ACSRC373
53	ACLN(LL,NC)=0.0	ACSRC374
131	ND=NC(JJ)	ACSRC375
	DC 54 I=1,NACTYP	ACSPC376
	DEP=TDPPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)	ACSRC377
	IF(DEP.LE.0.0) GO TO 54	ACSRC378
	AA=ENGNO(I,1)	ACSPC379
	IF(IEGFLG.GT.0) AA=ENGNO(I,2)	ACSPC380
	TIME=ACLN(11,ND)/TXISPD(I)	ACSRC381
	FRC=AA*DEP*TIME*FRAC(I)	ACSRC382
	DO 55 L=1,NPLTS	ACSRC383
	KK=L+12	ACSRC384
55	ACLN(KK,ND)=ACLN(KK,ND)+FRC*ACEMFC(I,2,L)	ACSRC385
54	CCONTINUE	ACSRC386
52	CONTINUE	ACSRC387
C		ACSRC388
C	END TAXIWAY SEGMENT LOOP	ACSRC389
C		ACSRC390
C		ACSRC391
C	DETERMINE AIRCRAFT OUTBOUND PARKING AREA ASSOCIATED	ACSPC392
C	WITH TAXIWAY PATH	ACSRC393
C		ACSRC394
	DO 56 I=1,NPKSRC	ACSRC395
	II=I	ACSRC396
	IDPK=IACAR(1,I)	ACSRC397
	IF(IDPK.EQ.IDOBPA(J,N)) GO TO 58	ACSPC398
56	CONTINUE	ACSRC399
	PRINT 57, IDOBPA(J,N), J,N	ACSRC400
57	FORMAT(22HOUTBOUND PARKING AREA, I3, 11H OF TAXIWAY, I3, 8H, RUNWAY,	ACSRC401
	. I3, 47H IS NOT CONSISTENT WITH PARKING AREA ID NUMBERS)	ACSRC402
	STOP	ACSRC403
C		ACSRC404
C	ALLOCATE ALL AIRCRAFT IDLE AT STARTUP, DEPARTURE FUEL	ACSRC405
C	VENTING AND SERVICE VEHICLE EMISSIONS TO APPROPRIATE AREA	ACSRC406
C		ACSRC407
58	NSQ=IACAR(2,II)	ACSRC408
	DO 59 I=1,NACTYP	ACSRC409
	DEP=TDPPFR(J,I,N)*DEPFCN(23,I,N)*ANNDEP(I)	ACSRC410
	IF(DEP.EQ.0.0) GO TO 59	ACSRC411
	AA=ENGNO(I,1)	ACSRC412
	IF(IEGFLG.GT.0) AA=ENGNO(I,2)	ACSRC413
	TIME=SRTUPT(I)/60.	ACSRC414
	FRC=AA*DEP*TIME*FRAC(I)	ACSPC415
	TVP=EXP(ALPHA(JES1(I))-BETA(JES1(I)))/TEMK)	ACSRC416
	DC 60 L=1,NSQ	ACSPC417
	JJ=II+L-1	ACSRC418
	DO 61 K=1,NPLTS	ACSRC419
	KK=K+5	ACSRC420
	ACAR(KK,JJ)=ACAR(KK,JJ)+((FRC*ACEMFC(I,1,K))+	ACSRC421
	. ((DPSVEM(K,I,1)+DPSVEM(K,I,2)+DPSVEM(K,I,3)+DPSVEM(K,I,4)	ACSRC422
	. +DPSVEM(K,I,5))*DEP*FRAC(I))*PARFCT(JJ)	ACSRC423
	IF(K.EQ.2) ACAR(KK,JJ)=ACAR(KK,JJ)+DPFLVT(I)*DEP*FLDENS(	ACSRC424
	. JES1(I))*FRAC(I)*PARFCT(JJ)	ACSRC425
61	CCONTINUE	ACSRC426
60	CCONTINUE	ACSRC427
59	CCONTINUE	ACSRC428
51	CONTINUE	ACSRC429
C		ACSPC430
C	END OUTBOUND TAXIWAY LOOP	ACSRC431
C		ACSPC432
		ACSPC433

	NB=NB+1	ACSRC434
C		ACSRC435
C	SET UP AREA SOURCE AT TAIL OF RUNWAY AND ALLOCATE	ACSRC436
C	ENGINE CHECK EMISSIONS TO IT	ACSRC437
C		ACSRC438
	ACAR(1,NB)=RNWY(2,N)-.05 * SIN(THETA)	ACSRC439
	ACAR(2,NB)=RNWY(3,N)-.05*COS(THETA)	ACSRC440
	ACAR(3,NB)= ACLNDZ/2.	ACSRC441
	ACAR(4,NB)= 100.0	ACSRC442
	ACAR(5,NB)= ACLNDZ	ACSRC443
	DO 65 K=1,NPLTS	ACSRC444
	KK=K+5	ACSRC445
65	ACAR(KK,NB)=0.0	ACSRC446
	DO 66 I=1,NACTYP	ACSRC447
	DEP=DEPFCN(23,I,N)*ANNDEP(I)	ACSRC448
	IF(DEP.EQ.0.0) GO TO 66	ACSRC449
	AA=ENGNO(I,1)	ACSRC450
	IF(IEGFLG.GT.0) AA=ENGNO(I,2)	ACSRC451
	TIME= EGCHKT(I)/60.	ACSRC452
	FRC= TIME *DEP*AA*FRAC(I)	ACSRC453
	DC 67 K=1,NELTS	ACSRC454
	KK=K+5	ACSRC455
67	ACAR(KK,NB)=ACAR(KK,NB)+ FRC* ACEMFC(I,3,K)	ACSRC456
66	CONTINUE	ACSRC457
10	CONTINUE	ACSRC458
C		ACSRC459
C	END RUNWAY LOOP	ACSRC460
C		ACSRC461
	NACAR=NB	ACSRC462
	NC1=NC	ACSRC463
69	NC=NC1	ACSRC464
C		ACSRC465
C	BEGIN LOOP OVER N RUNWAYS	ACSRC466
C		ACSRC467
	DO 79 N=1,NRWYS	ACSRC468
C		ACSRC469
C	IS RUNWAY USED WITH THIS WIND DIRECTION?	ACSRC470
C		ACSRC471
	IF(IUSWD(IWD,N).EQ.0) GO TO 79	ACSRC472
C		ACSRC473
C	BEGIN LOOP OVER I AIRCRAFT USED	ACSRC474
C		ACSRC475
	DO 70 I=1,NACTYP	ACSRC476
C		ACSRC477
C	CALCULATE RUNWAY DEPARTURES FOR EACH AIRCRAFT TYPE	ACSRC478
C		ACSRC479
	DEP=DEPFCN(23,I,N)*ANNDEP(I)	ACSRC480
C		ACSRC481
C	ANY AIRCRAFT DEPARTING FROM THIS RUNWAY?	ACSRC482
C		ACSRC483
	IF(DEP.EQ.0.0) GO TO 70	ACSRC484
C		ACSRC485
C	CALL DEPART TO CALCULATE POINTS IN TAKEOFF PATH ACCORDING	ACSRC486
C	TO CURRENT METEOROLOGICAL CONDITIONS	ACSRC487
C		ACSRC488
	CALL DEPART (N,I)	ACSRC489
	AA=ENGNO(I,1)	ACSRC490
C		ACSRC491
C	SET UP LINE SOURCES FOR RUNWAY ROLL AND CLIMBOUT MODES 1 AND 2	ACSRC492
C	AND ALLOCATE POLLUTANT EMISSIONS	ACSRC493
C		ACSRC494
	DO 71 J=1,3	ACSRC495



DO 72 K=1,3	ACSRC496
KK=K+NC	ACSRC497
JK=6*K-6+J	ACSRC498
ACLN(J, KK) = DEPPCN(JK, I, N)	ACSRC499
72 ACLN(J+5, KK) = DEPPCN(JK+6, I, N)	ACSRC500
JJ=NC+J	ACSRC501
JK=6*J-2	ACSRC502
ACLN(4, JJ) = ARRPCN(24, I, N)	ACSRC503
ACLN(5, JJ) = DEPPCN(24, I, N)	ACSRC504
ACLN(09, JJ) = DEPPCN(JK, I, N)	ACSRC505
ACLN(10, JJ) = DEPPCN(JK+6, I, N)	ACSRC506
ACLN(11, JJ) = DEPPCN(JK+1, I, N)	ACSRC507
ACLN(12, JJ) = DEPPCN(JK+2, I, N)	ACSRC508
JMODE=J+3	ACSRC509
DO 73 K=1, NPLTS	ACSRC510
KK=K+12	ACSRC511
ACLN(KK, JJ) = AA*ACEMFC(I, JMODE, K) *DEP*DEPPCN(JK+2, I, N) *FRAC(I)	ACSRC512
73 CONTINUE	ACSRC513
71 CONTINUE	ACSRC514
NC=NC+3	ACSRC515
70 CONTINUE	ACSRC516
C	ACSRC517
C END AIRCRAFT LOOP	ACSRC518
C	ACSRC519
79 CONTINUE	ACSRC520
C	ACSRC521
C END RUNWAY LOOP	ACSRC522
C	ACSRC523
NACLN=NC	ACSRC524
RETURN	ACSRC525
END	ACSRC526

## FUNCTION AINE

### Purpose:

1. To translate the line and receptor coordinates to an x-axis along the wind vector, placing the origin of the line at its low end.
2. To set up the necessary parameters for the CAVL and QMOD routines.
3. To determine the concentration due to the given line.

### Input:

The current wind direction and speed, and the receptor and line source data.

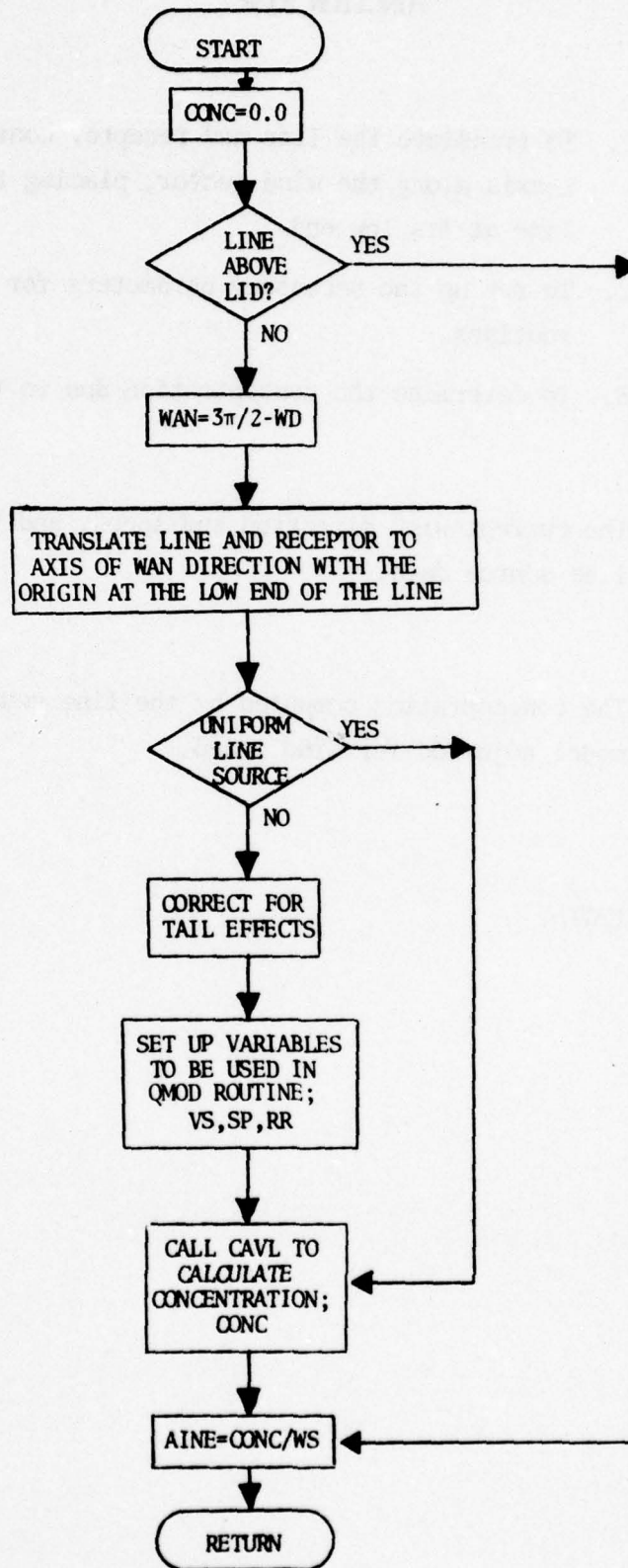
### Output:

The concentration computed by the line source diffusion model adjusted for wind speed.

### Subroutines Called:

CAVL

FUNCTION AINE(WD)





C	FUNCTION AINE (WD)	AINEO000
C		AINEO001
C	THIS FUNCTION TRANSLATES THE LINE AND RECEPTOR COORDINATES TO AN	AINEO002
C	X-AXIS ALONG THE WIND VECTOR, PLACING THE ORIGIN OF THE LINE AT	AINEO003
C	ITS LOW END. THE VEHICLE MOVES FROM (X1,Y1,Z1) TO (X2,Y2,Z2)	AINEO004
C		AINEO005
	COMMON /MFT/ WS,WSMPH,IWS,WX,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,	AINEO006
	. TENK	AINEO007
	COMMON /RCPT/ NRECEP,RECEP(2,312)	AINEO008
	COMMON /INFO/ IRECEP,IWDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,	AINEO009
	. V1,V2,DL,TIME,EMIS(6),NPOL	AINEO010
	COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,	AINEO011
	. WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND	AINEO012
	DATA PI32/4.7123890/	AINEO013
C		AINEO014
	CCNC = 0.	AINEO015
C		AINEO016
C	IF LINE IS ABOVE LID, DO NOT CALCULATE CONC	AINEO017
C		AINEO018
	IF(ZW1.GE.HLID-.5) GO TO 60	AINEO019
C		AINEO020
C	TRANSLATE LINE AND RECEPTOR TO AXIS OF WAN DIRECTION	AINEO021
C		AINEO022
	WAN=PI32-WD	AINEO023
	CSAN=COS(WAN)	AINEO024
	SNAN=SIN(WAN)	AINEO025
	XW2=(X2-X1)*CSAN+(Y2-Y1)*SNAN	AINEO026
	YW2=(X1-X2)*SNAN+(Y2-Y1)*CSAN	AINEO027
	XR = RECEP(1,IRECEP) * 1000.	AINEO028
	YR = RECEP(2,IRECEP) * 1000.	AINEO029
	ZST=ZW1	AINEO030
	ZND=ZW2	AINEO031
	IF(Z1.LE.Z2) GO TO 5	AINEO032
	XW2=-XW2	AINEO033
	XST=XW2	AINEO034
	YW2=-YW2	AINEO035
	YST=YW2	AINEO036
	XND=0.0	AINEO037
	YND=0.0	AINEO038
	XRCP=(XR-X2)*CSAN+(YR-Y2)*SNAN	AINEO039
	YRCP=(X2-XR)*SNAN+(YR-Y2)*CSAN	AINEO040
	GO TO 8	AINEO041
5	CCNTINUE	AINEO042
	XST=0.0	AINEO043
	YST=0.0	AINEO044
	XND=XW2	AINEO045
	YND=YW2	AINEO046
	XRCP=(XR-X1)*CSAN+(YR-Y1)*SNAN	AINEO047
	YRCP=(X1-XR)*SNAN+(YR-Y1)*CSAN	AINEO048
8	CONTINUE	AINEO049
	ZRCP = 2.	AINEO050
C		AINEO051
C	IS THIS A UNIFORM LINE SOURCE	AINEO052
C		AINEO053
	50 IF(IAD.EQ.0) GO TO 500	AINEO054
C		AINEO055
C	CORRECT FOR TAIL EFFECTS IF ARRIVAL OR DEPARTURE	AINEO056
C		AINEO057
	CSA = -XW2 / DL	AINEO058
	WSC = 2 * WS * CSA	AINEO059
	EXT = TAIL / DL	AINEO060
	DX = XW2 * EXT	AINEO061

DY = YW2 * EXT	AINF0062
XW2 = XW2 + DX	AINE0063
YW2 = YW2 + DY	AINE0064
VS = TAIL / TIME	AINE0065
W1 = V1 + VS	AINE0066
W2 = V2 + VS	AINE0067
YY1 = SQRT(W2 + W1 * (W1 + WSC))	AINE0068
YY2 = SQRT(W2 + W2 * (W2 + WSC))	AINE0069
SP = YY2	AINE0070
ARG = (YY2 + W2 + WSC/2.) / (YY1 + W1 + WSC/2.)	AINE0071
G = YY2 - YY1 - WSC/2. * ALOG(ARG)	AINE0072
RR = A / G	AINE0073
IF (Z1.NE.Z2.AND.IAD.EQ.1) GO TO 500	AINE0074
XRCP = XRCP + DX	AINE0075
YRCP = YRCP + DY	AINE0076
C	AINE0077
C CALCULATE THE CONCENTRATION DUE TO THIS LINE	AINE0078
C	AINE0079
500 CONC=CAVL(XRCP,YRCP,ZRCP)	AINE0080
60 AINE = CONC / WS	AINE0081
RETURN	AINE0082
END	AINE0083

## BLOCK DATA

Purpose:

To initialize data in common blocks.

Input:

None

Output:

None



	BLOCK DATA	BLKDT000
C		BLKDT001
C	INITIALIZE DATA IN COMMON BLOCKS FOR SHORT TERM MODEL	BLKDT002
C		BLKDT003
	REAL*8 POLNAM	BLKDT004
C		BLKDT005
	CCHMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR	BLKDT006
	COMMON /DEFAULT/ ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	BLKDT007
	CCHMON /DSTRET/ ACHO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),	BLKDT008
	. VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),	BLKDT009
	. CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1	BLKDT010
	CCHMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,B,V12,VS,	BLKDT011
	. WS2,WSC,RR,SP,AA1,AA2,AA3,AA4,AA5,AA6	BLKDT012
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE	BLKDT013
	COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,	BLKDT014
	. NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),	BLKDT015
	. ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)	BLKDT016
	COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,IF	BLKDT017
	CCHMON /WNDPRO/ XP(6)	BLKDT018
C		BLKDT019
C	*****DATA STATEMENTS*****	BLKDT020
C		BLKDT021
	DATA XW1, YW1, TAIL / 0.0, 0.0, 140. /	BLKDT022
	DATA XP / 0.2, 0.2, 0.2, 0.3, 0.4, 0.4 /	BLKDT023
	DATA ALPHA / 11.70365, 11.10675, 12.42382, 12.68789, 13.687,	BLKDT024
	. 13.038, 13.024 /	BLKDT025
	DATA BETA / 2868.54, 3129.5187, 3276.8848, 5108.4194,5329.139,	BLKDT026
	. 4789.301, 4782.209 /	BLKDT027
	DATA FLDENS / 0.695, 0.773, 0.693, 0.842, 0.824, 0.807, 0.807 /	BLKDT028
	DATA ACLNDY, ACLNDZ / 20.0,8.0 /	BLKDT029
	DATA ITAPE, IONCE / 21, 0 /	BLKDT030
	DATA POLNAM / 8H CO ,8H HC ,8H NOX ,8H PT ,	BLKDT031
	. 8H SO2 ,8H POL6 /	BLKDT032
	DATA ENPT,ENAR,ENLN,ABPT,ABAR,ABLN,ACPT,ACAR,ACLN / 12660*0.0 /	BLKDT033
	DATA NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,NACAR,NACLN /9*0.0/	BLKDT034
	END	BLKDT035

## FUNCTION CAVL

### Purpose:

To compute the coupling coefficient at a receptor due to a line source.

### Input:

Meteorological conditions: wind speed; stability; mixing height; critical distance for vertical mixing; psuedo downwind distances for horizontal and vertical spreads of the line source.

Source parameters: end point coordinates of the line (X-axis has been chosen to be along the wind vector); IAD flag for uniform or non-uniform line.

Receptor coordinates.

### Output:

CAVL, the coupling coefficient.

### Procedure:

1. Test whether the receptor is located with respect to the line source such that the concentration is negligible.
2. If the angle between the wind vector and line is sufficiently small, and the line is sufficiently long, set a flag for the line to be segmented.
3. Compute effective downwind distance and the horizontal and vertical dispersion coefficients.
4. Determine factor to be used in subdividing the line.
5. Test whether the line has a uniform density. If it is a runway used for aircraft arrival or departure (non-uniform density), call subroutine QMOD.
6. Determine the proper expression to be used and compute the concentration due to the line segment.
7. Test whether further segments need be considered. If not, output the concentration for the given receptor.

**Functions  
Called:**

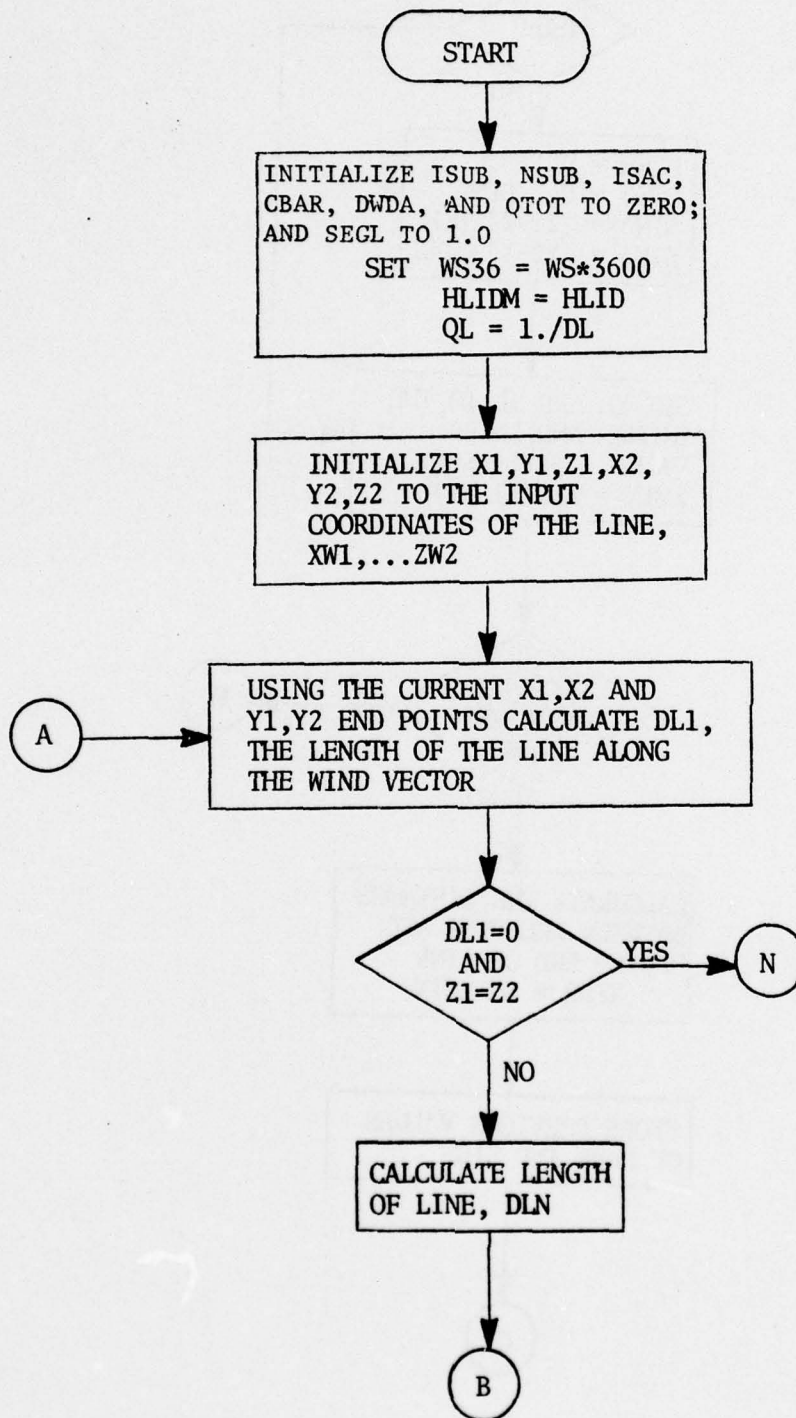
SIGY,SIGZ,DIFERF

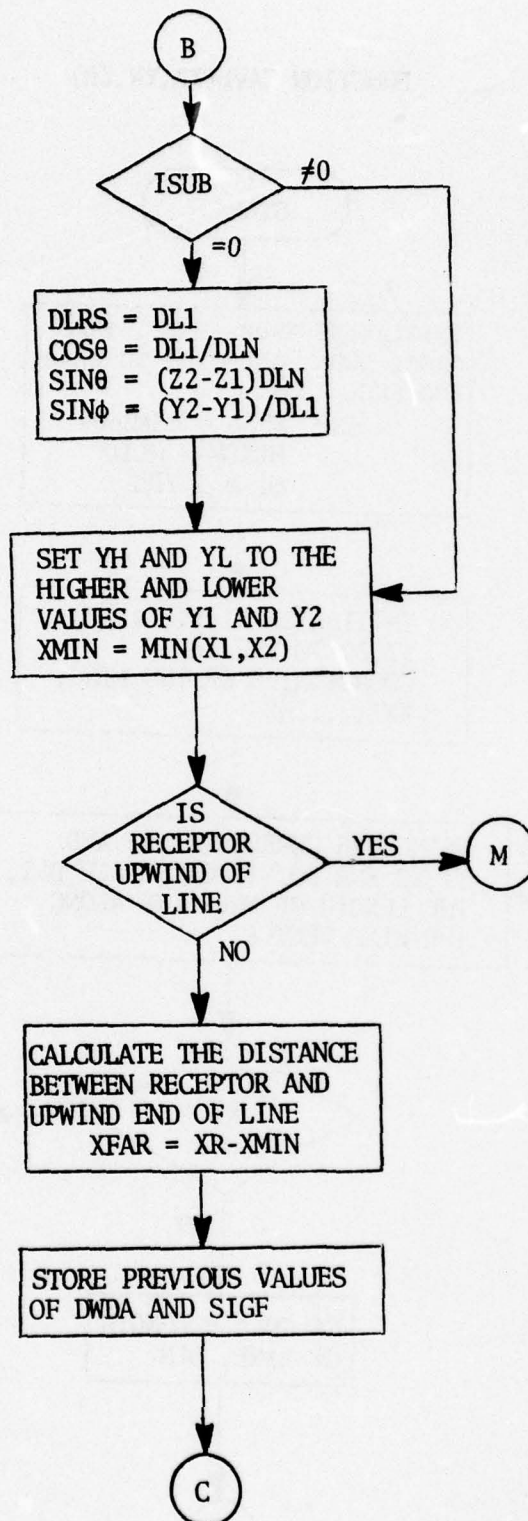
**Subroutine  
Called:**

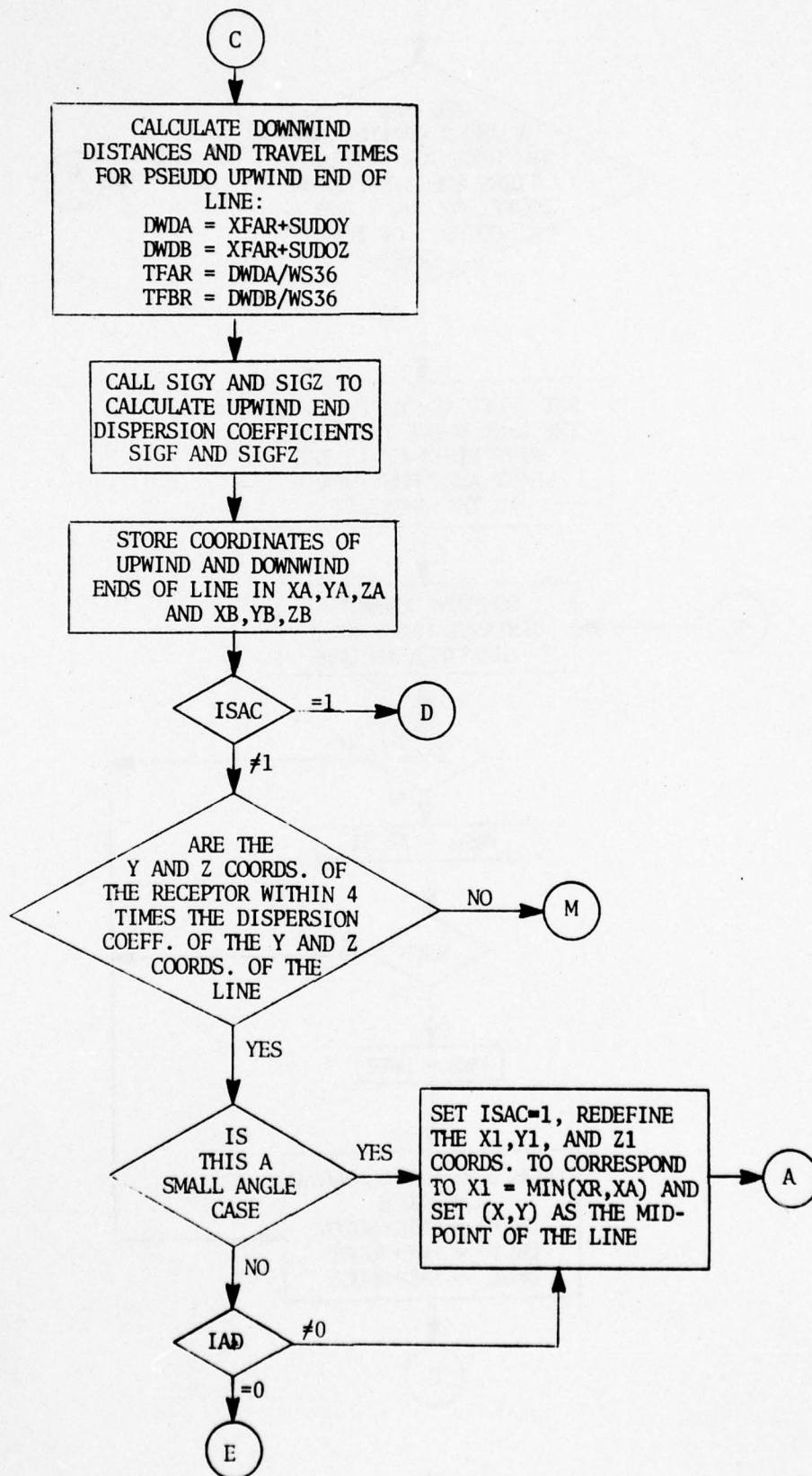
QMOD



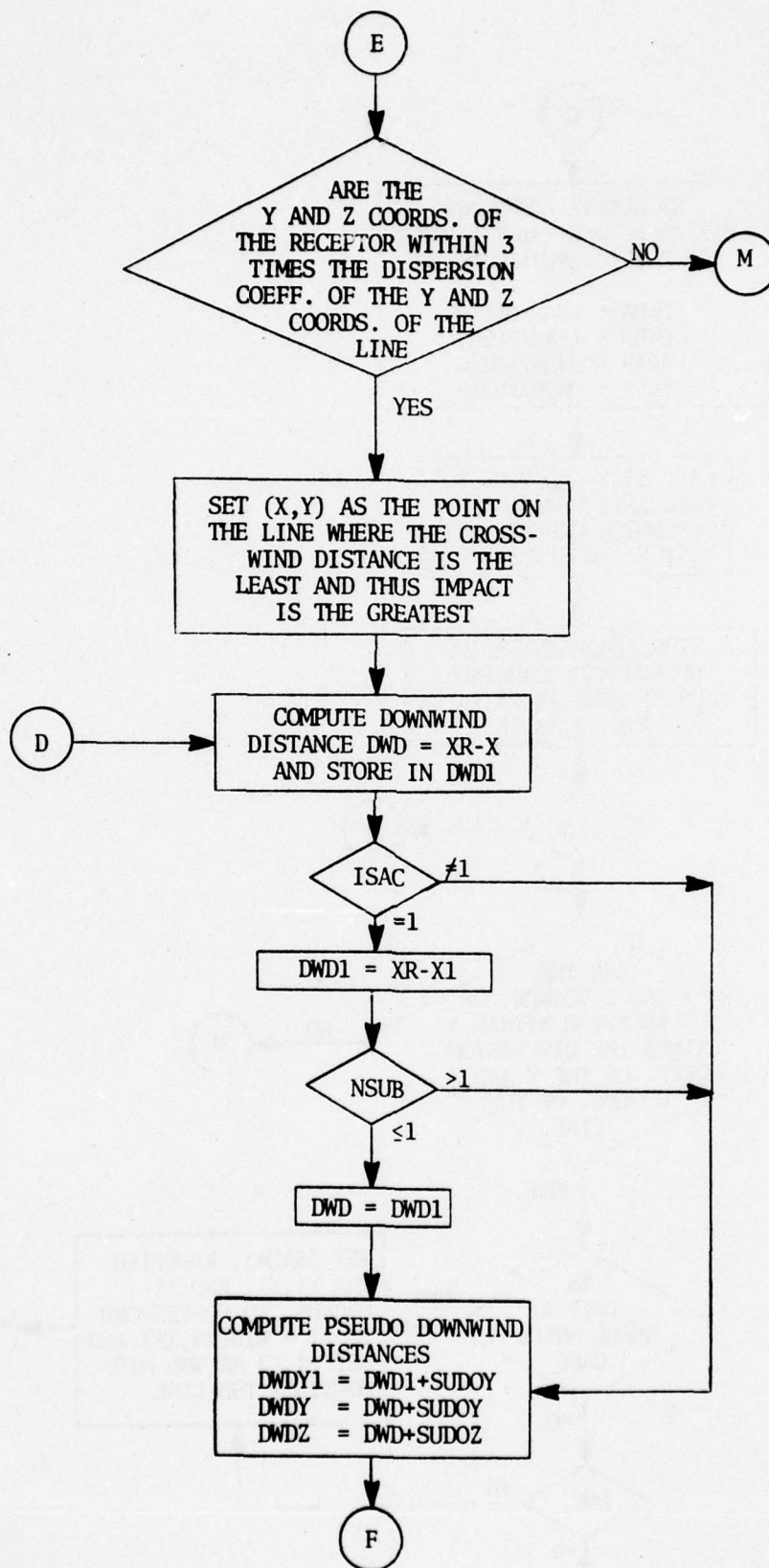
FUNCTION CAVL(XR,YR,ZR)

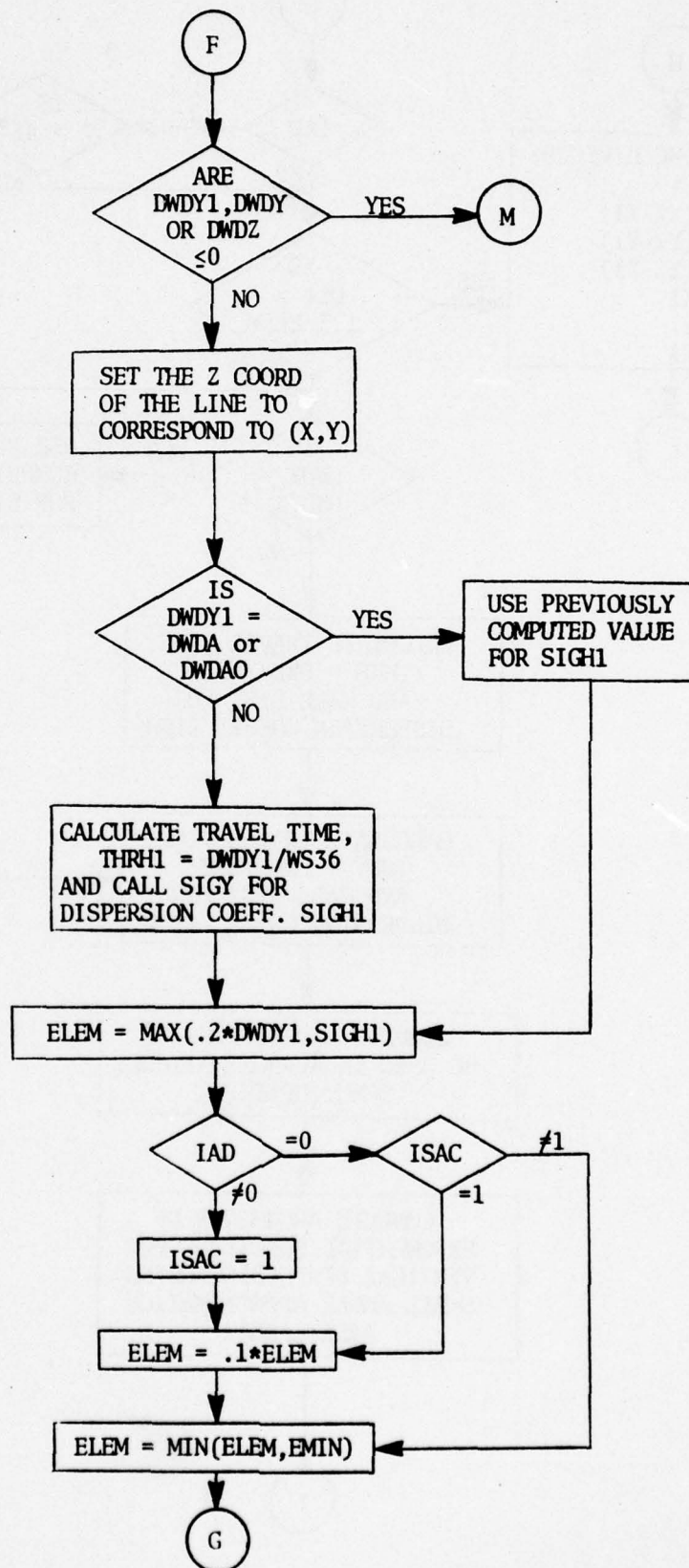


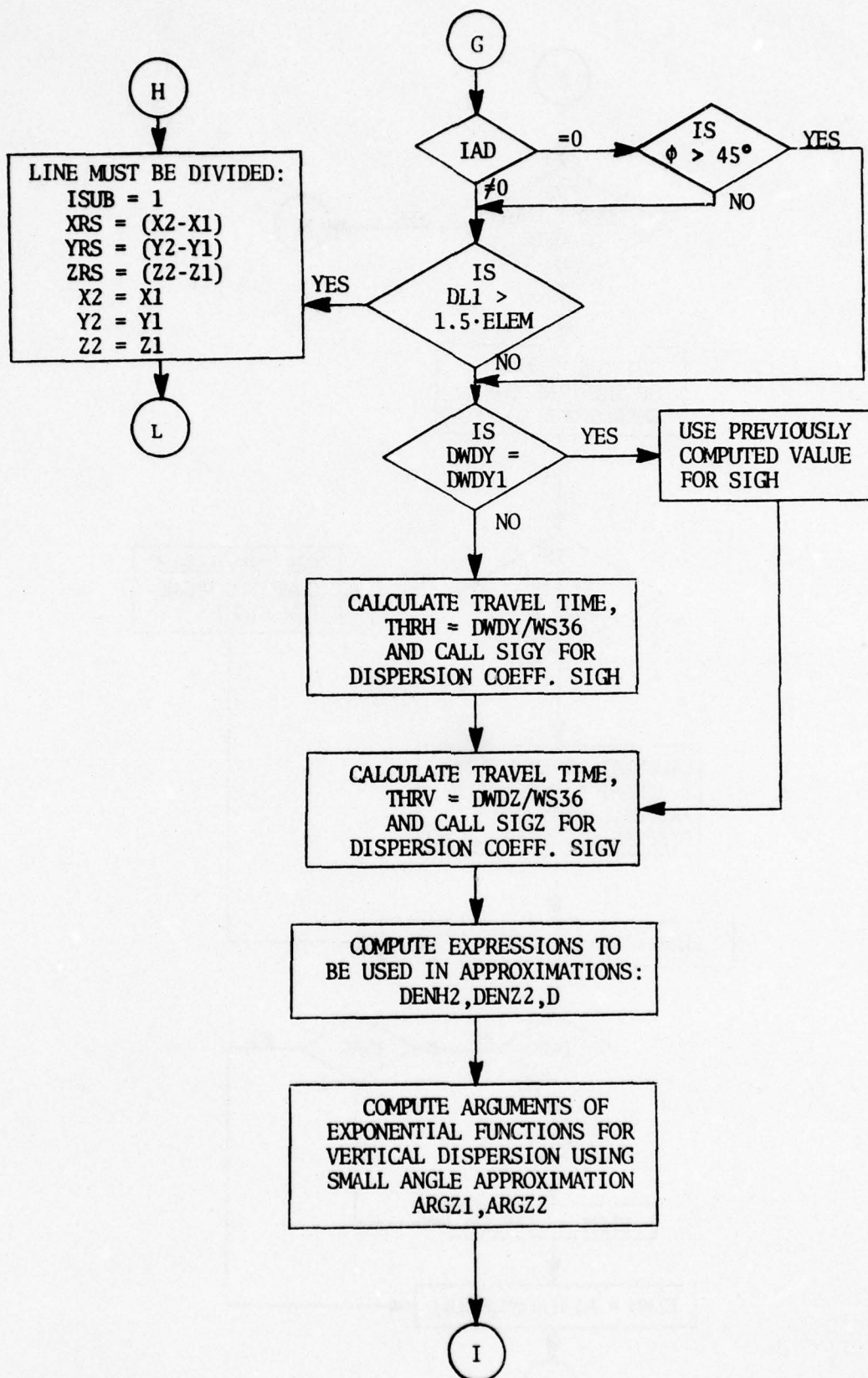




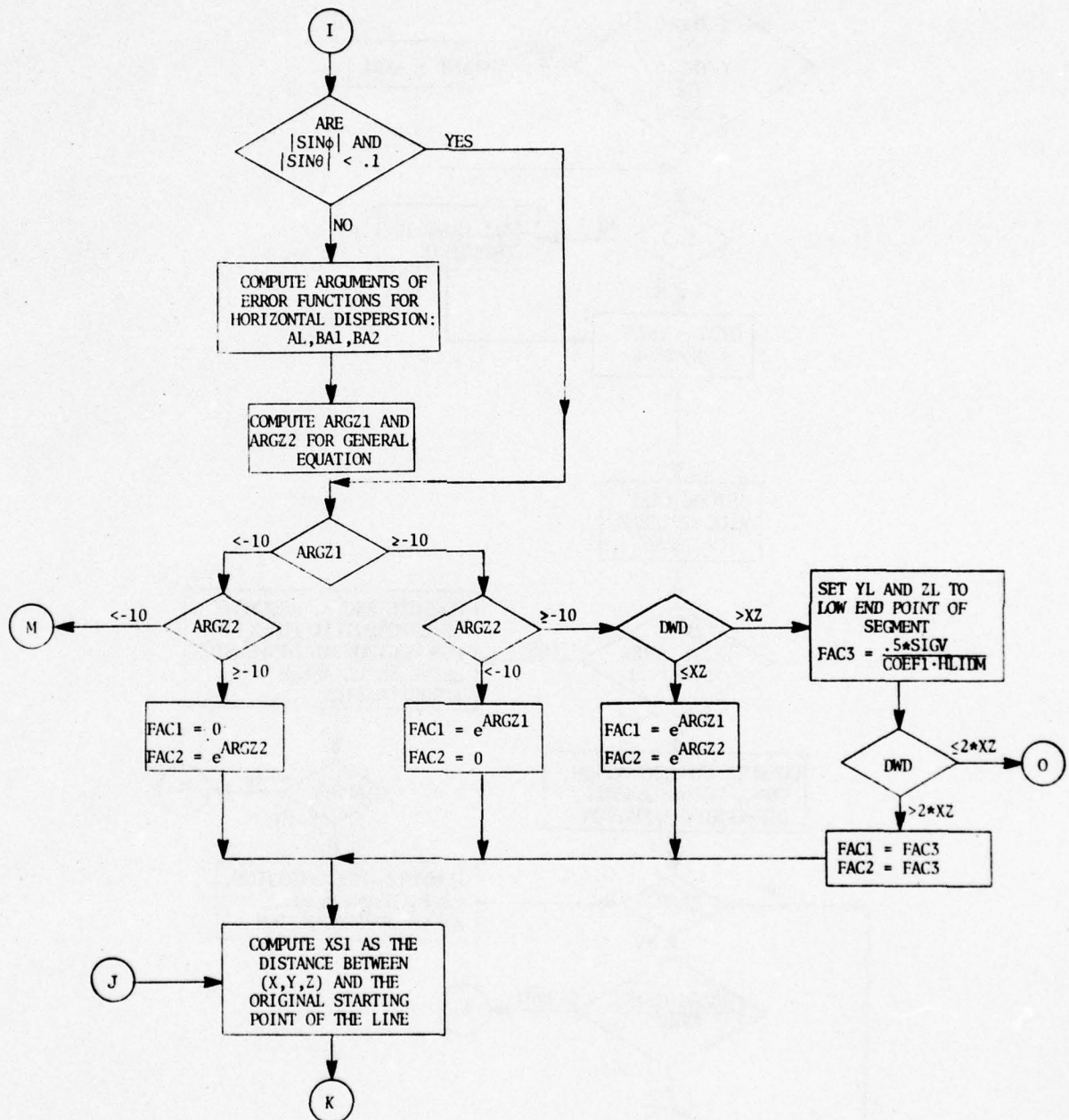


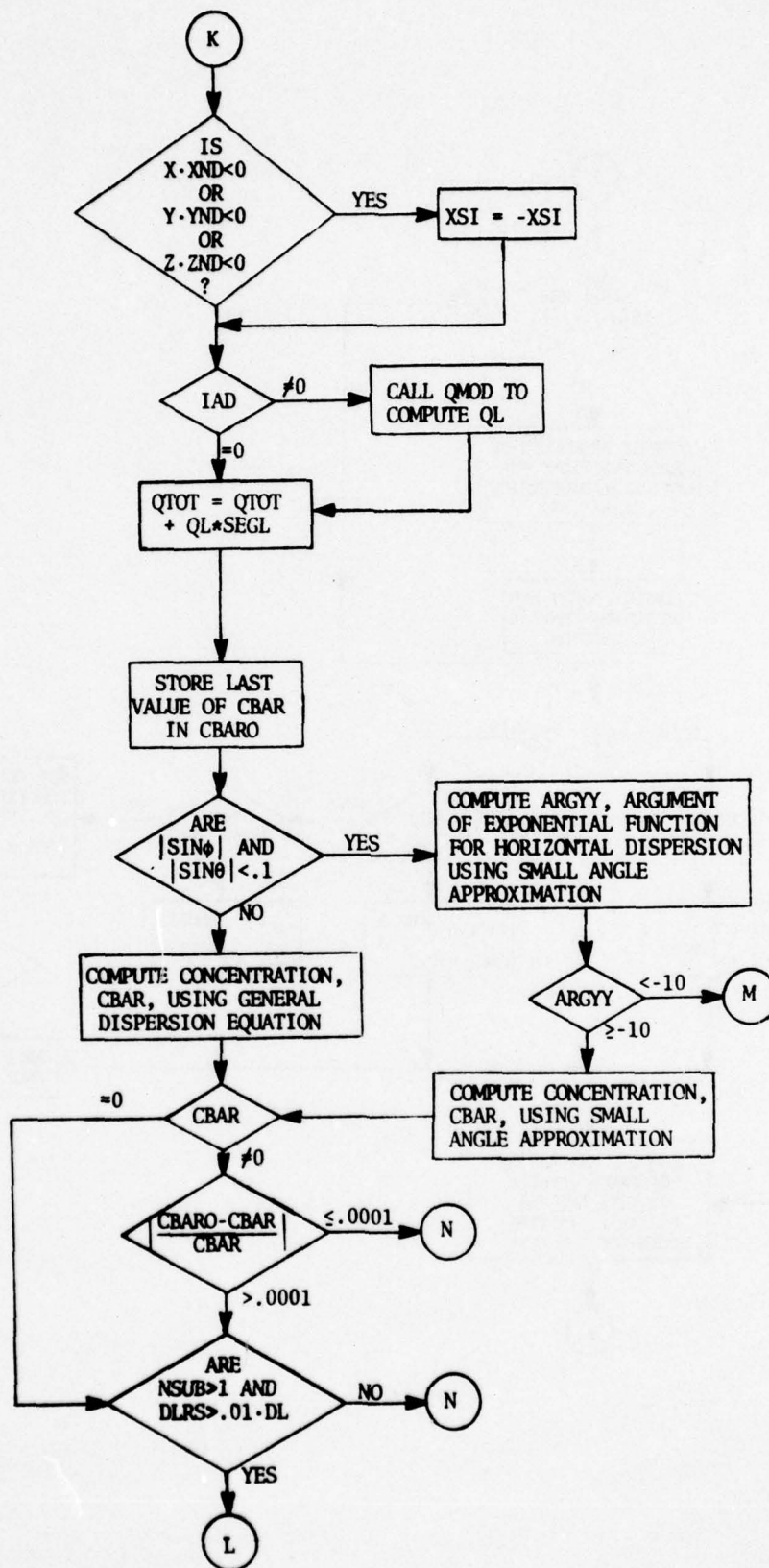


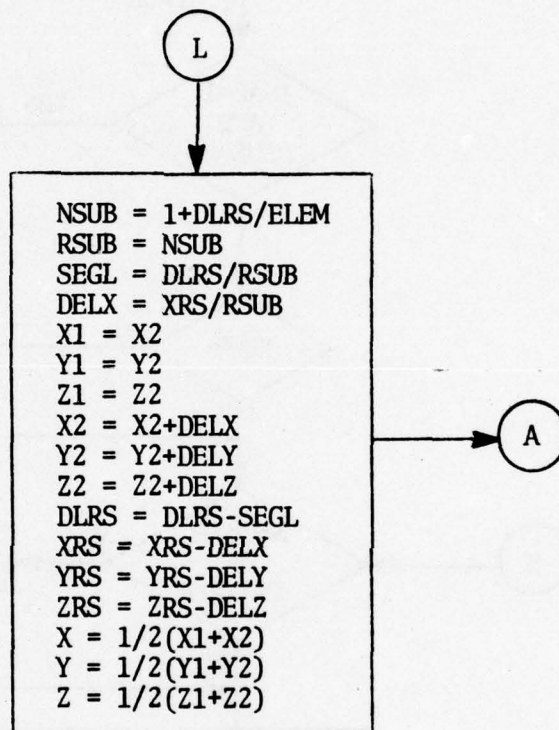




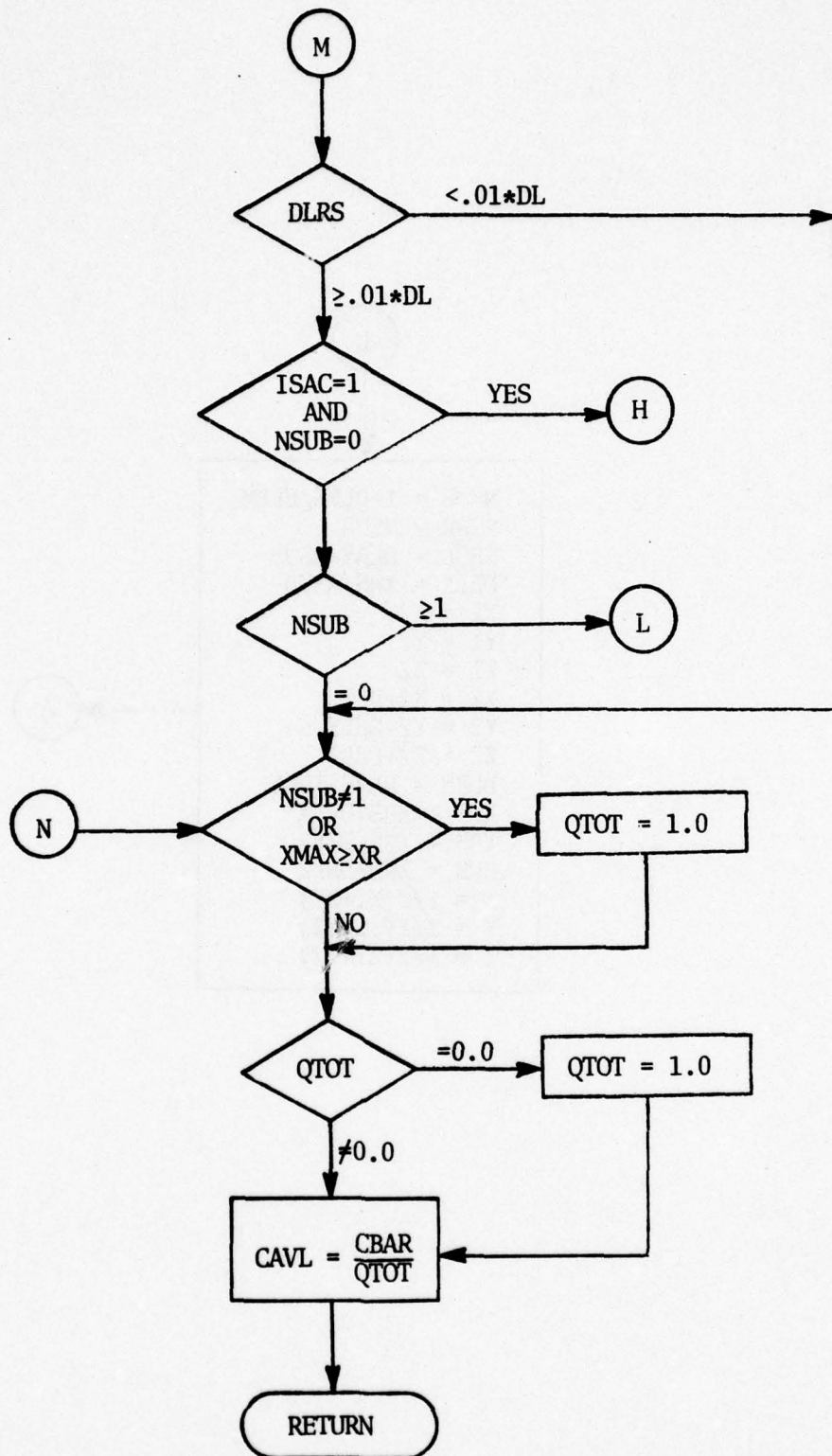


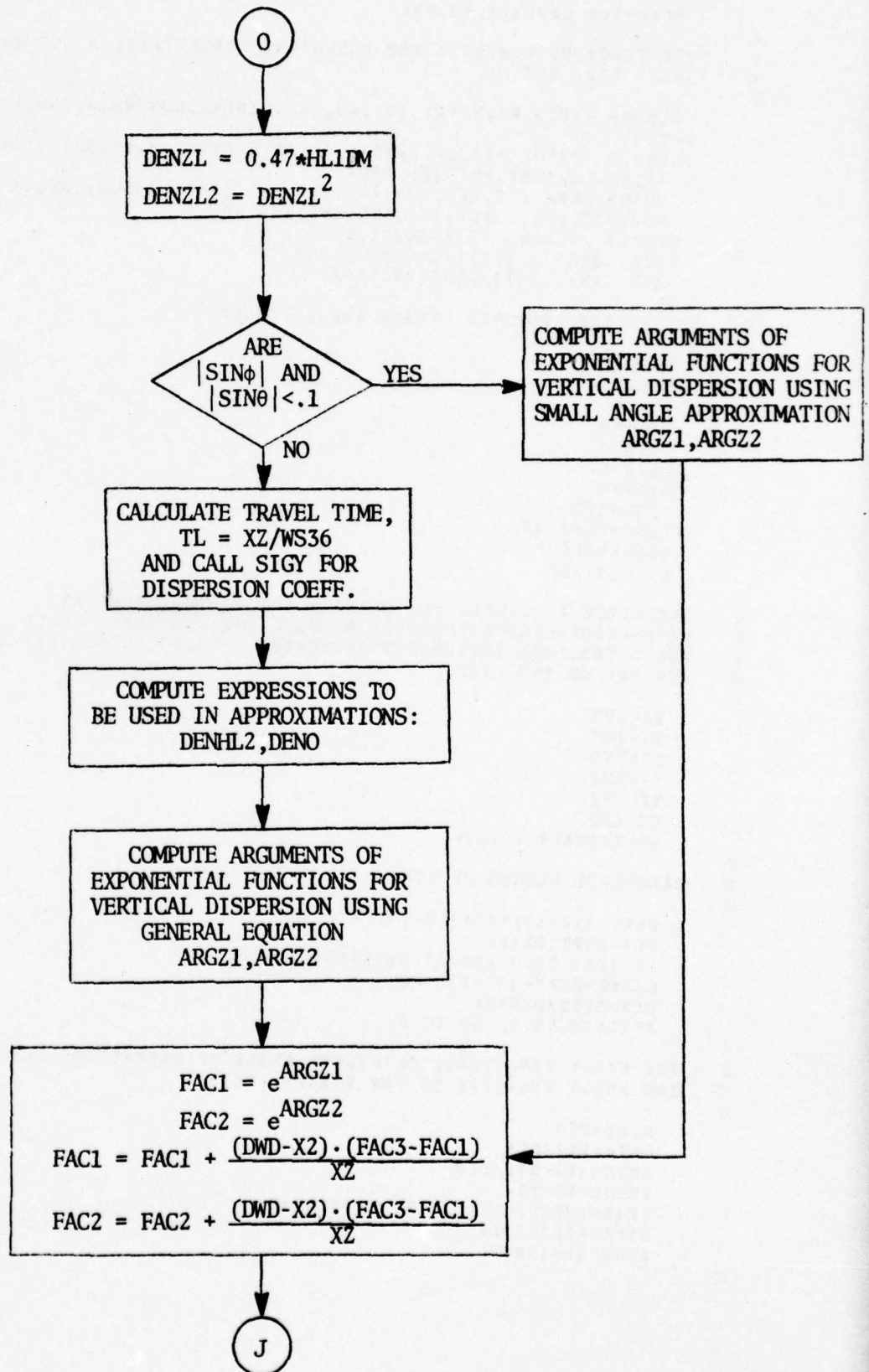












C	FUNCTION CAVL (XR,YR,ZR)	CAVL0000
C	THIS FUNCTION COMPUTES THE POLLUTANT CONCENTRATION DUE TO A	CAVL0001
C	FINITE LINE SOURCE	CAVL0002
C		CAVL0003
C	COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,	CAVL0004
	. TEMK	CAVL0005
	COMMON /INFO/ IRECEP,IWDIR,ITYPE,HTAERO,X5,Y5,Z5,W,DELZ,X6,Y6,Z6,	CAVL0006
	. V1,V2,DL,TIME,EMIS(6),NPOL	CAVL0007
	COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDCY,SUDOZ,IAD,TAIL,B,V12,VS,	CAVL0008
	. WS2,WSC,RR,SP,XST,YST,ZST,XND,YND,ZND	CAVL0009
	COMMON /XTRAN/ XZ,WSMD,TY,TZ	CAVL0010
	DATA COEF1 /.39894/,COEF2 /.31831/	CAVL0011
	DATA CAN/0.7071/,EMIN/9.144/	CAVL0012
C		CAVL0013
C	INITIALIZE COUNTERS, FLAGS AND VARIABLES	CAVL0014
C		CAVL0015
C	ISUB=0	CAVL0016
	NSUB=0	CAVL0017
	ISAC=0	CAVL0018
	LSAC=0	CAVL0019
	CEAR=0.	CAVL0020
	DWDA=0.	CAVL0021
	QTOT=0.	CAVL0022
	SEGL=1.0	CAVL0023
	WS36=WS*3600.	CAVL0024
	HLIDM=HLID	CAVL0025
	QL = 1./DL	CAVL0026
C		CAVL0027
C	INTRODUCE A GENERAL SET OF NOTATION SO THAT THE SAME	CAVL0028
C	DISPERSSION CALCULATION CAN BE USED FOR THE SMALL ANGLE CASE	CAVL0029
C	WHERE THE LINE IS FURTHER SEGMENTED. X1,Y1,Z1 NOW REFER TO THE	CAVL0030
C	LOW END OF THE LINE.	CAVL0031
C		CAVL0032
	X1=XW1	CAVL0033
	Y1=YW1	CAVL0034
	Z1=ZW1	CAVL0035
	X2=XW2	CAVL0036
	Y2=YW2	CAVL0037
	Z2=ZW2	CAVL0038
	XMAX=AMAX1(X1,X2)	CAVL0039
C		CAVL0040
C	CALCULATE LENGTH OF LINE	CAVL0041
C		CAVL0042
C	5 DLXY=(X2-X1)**2+(Y2-Y1)**2	CAVL0043
	DL1=SQRT(DLXY)	CAVL0044
	IF (DL1.EQ.0.AND.Z1.EQ.Z2) GO TO 600	CAVL0045
	DLXYZ=DLXY+(Z2-Z1)**2	CAVL0046
	DIN=SQRT(DLXYZ)	CAVL0047
	IF (ISUB.NE.0) GO TO 6	CAVL0048
C		CAVL0049
C	THE FIRST TIME THRU, CALCULATE ANGLE OF ELEVATION, THETA,	CAVL0050
C	AND ANGLE RELATIVE TO THE X-AXIS, PHI	CAVL0051
C		CAVL0052
	DLRS=DL1	CAVL0053
	CSTH=DL1/DLN	CAVL0054
	SNTH=(Z2-Z1)/DLN	CAVL0055
	PROJL=Y2-Y1	CAVL0056
	IF (ABS(PROJL).LT.1.E-20) PROJL=0.	CAVL0057
	SNFI=PROJL/DL1	CAVL0058
	ASNFI=ABS(SNFI)	CAVL0059
C		CAVL0060
		CAVL0061



C	FIND HIGH AND LOW ENDS OF LINE AS PROJECTED ON THE X-Y PLANE	CAVL0062
C		CAVL0063
	6 CONTINUE	CAVL0064
	IF(Y1.GT.Y2) GO TO 1	CAVL0065
	YH=Y2	CAVL0066
	YI=Y1	CAVL0067
	GC TO 2	CAVL0068
	1 YH=Y1	CAVL0069
	YI=Y2	CAVL0070
	2 CONTINUE	CAVL0071
C		CAVL0072
C	TEST THE RECEPTOR LOCATION RELATIVE TO THE LINE SOURCE	CAVL0073
C		CAVL0074
	XMIN=AMIN1(X1,X2)	CAVL0075
	IF ((XMIN-XR).GE.0.5) GO TO 500	CAVL0076
C		CAVL0077
C	RECEPTOR IS DOWNWIND, FIND DISTANCE TO UPWIND END OF LINE	CAVL0078
C		CAVL0079
	XFAF=XR-XMIN	CAVL0080
C		CAVL0081
C	STOPP PPREVIOUS VALUES AND COMPUTE NEW DOWNWIND DISTANCES	CAVL0082
C	AND TRAVEL TIMES FOR PSEUDO UPWIND END OF LINE	CAVL0083
C		CAVL0084
	DWDAC=DWDA	CAVL0085
	SIGFO=SIGF	CAVL0086
	DWDA=XFAF+SUDCY	CAVL0087
	DWDE=XFAP+SUDCZ	CAVL0088
	TFAF=DWDA/WS36	CAVL0089
	TFBR=DWDB/WS36	CAVL0090
C		CAVL0091
C	COMPUTE UPWIND END DISPERSION COEFFICIENTS	CAVL0092
C		CAVL0093
	SIGF=SIGY(JSTAB,TFAR)	CAVL0094
	SIGFZ=SIGZ(JSTAB,TFBR)	CAVL0095
C		CAVL0096
C	STOPP LINE COORDINATES	CAVL0097
C		CAVL0098
	IF(X1.LE.X2) GO TO 21	CAVL0099
	XA=X2	CAVL0100
	YA=Y2	CAVL0101
	XB=X1	CAVL0102
	YB=Y1	CAVL0103
	ZF=Z1	CAVL0104
	GC TO 22	CAVL0105
21	XA=X1	CAVL0106
	YA=Y1	CAVL0107
	XB=X2	CAVL0108
	YB=Y2	CAVL0109
	ZF=Z2	CAVL0110
22	CONTINUE	CAVL0111
	IF(ISAC.EQ.1) GO TO 4	CAVL0112
C		CAVL0113
C	ARE Y AND Z COORDS OF RECEPTOR WITHIN 4 TIMES THE DISPERSION	CAVL0114
C	COEFFICIENT OF THE Y AND Z COORDS OF THE LINE	CAVL0115
C		CAVL0116
	IF(YF.GT.(YH+4.*SIGF)) GO TO 500	CAVL0117
	IF(YF.LT.(YL-4.*SIGF)) GO TO 500	CAVL0118
	IF(ZF.GT.(Z2+4.*SIGFZ)) GO TO 500	CAVL0119
	IF(ZF.LT.(Z1-4.*SIGFZ)) GO TO 500	CAVL0120
	IF (ASNF .LT. CAN .AND. ABS(SNTH) .LT. CAN) GO TO 3	CAVL0121
	IF (IAD.NE.0) GO TO 3	CAVL0122
C		CAVL0123

C	ANGLE IS LARGE: ARE THE RECEPTOR COORDS WITHIN 3 TIMES THE	CAVLO124
C	DISPERSION COEFFICIENT OF THE LINE COORDS.	CAVLO125
C		CAVLO126
	IF (YR.GT. (YH+3.*SIGF)) GO TO 500	CAVLO127
	IF (YR.LT. (YL-3.*SIGF)) GO TO 500	CAVLO128
	IF (ZR.GT. (Z2+3.*SIGFZ)) GO TO 500	CAVLO129
	IF (ZR.LT. (Z1-3.*SIGFZ)) GO TO 500	CAVLO130
C		CAVLO131
C	SET (X,Y) AS POINT ON LINE WHERE IMPACT IS GREATEST	CAVLO132
C		CAVLO133
	$X=X1+(YR-Y1)*(X2-X1)/(Y2-Y1)$	CAVLO134
	IF (X.GT.XB) GO TO 333	CAVLO135
	IF (X.LT.XA) GO TO 33	CAVLO136
	Y=YF	CAVLO137
	GO TO 4	CAVLO138
C		CAVLO139
C	ANGLE IS SMALL: REDEFINE LINE COORDS AND SET (X,Y) AS	CAVLO140
C	MIDPOINT OF SEGMENT	CAVLO141
C		CAVLO142
	3 IF (ASN.F.LT.0.1.AND. (ABS(SNTH)).LT.0.1) LSAC=1	CAVLO143
	ISAC=1	CAVLO144
30	X=AMIN1(XF,XA)	CAVLO145
	$Y1=Y1+(X-X1)*(Y2-Y1)/(X2-X1)$	CAVLO146
	$Z1=Z1+(X-X1)*(Z2-Z1)/(X2-X1)$	CAVLO147
	X1=X	CAVLO148
	X2=XB	CAVLO149
	Y2=YB	CAVLO150
	Z2=ZB	CAVLO151
	$X=0.5*(X1+X2)$	CAVLO152
	$Y=0.5*(Y1+Y2)$	CAVLO153
	GO TO 5	CAVLO154
33	X=XA	CAVLO155
	Y=YA	CAVLO156
	GO TO 4	CAVLO157
333	X=XB	CAVLO158
	Y=YB	CAVLO159
C		CAVLO160
C	COMPUTE DOWNWIND DISTANCE	CAVLO161
C		CAVLO162
	4 DWD=XF-X	CAVLO163
	IF (DWD.LT.-.01) GO TO 30	CAVLO164
	DWD1=DWD	CAVLO165
	IF (ISAC.NE.1) GO TO 40	CAVLO166
	DWD1=XF-X1	CAVLO167
	IF (NSUB.LE.1) DWD=DWD1	CAVLO168
C		CAVLO169
C	COMPUTE PSEUDO DOWNWIND DISTANCES	CAVLO170
C		CAVLO171
40	DWDY1=DWD1+SUDOY	CAVLO172
	DWDY=DWD+SUDOY	CAVLO173
	DWDZ=DWD+SUDOZ	CAVLO174
C		CAVLO175
C	SET Z COORDINATE OF LINE	CAVLO176
C		CAVLO177
	IF (X1.EQ.X2) GO TO 44	CAVLO178
	$Z=Z1+(X-X1)*(Z2-Z1)/(X2-X1)$	CAVLO179
	GO TO 444	CAVLO180
44	$Z=Z1+(Y-Y1)*(Z2-Z1)/(Y2-Y1)$	CAVLO181
444	CONTINUE	CAVLO182
C		CAVLO183
C	COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR	CAVLO184
C	PSEUDO DOWNWIND DISTANCE	CAVLO185

C	IF (DWDY1.EQ.DWDA) GO TO 4111	CAVL0186
	IF (DWDY1.EQ.DWDAO) GO TO 4113	CAVL0187
	THRH=DWDY1/WS36	CAVL0188
	SIGH1=SIGY(JSTAB,THRH)	CAVL0189
4211	CONTINUE	CAVL0190
C		CAVL0191
C	DETERMINE FACTOR TO BE USED IN SUB-DIVIDING THE LINE	CAVL0192
C		CAVL0193
	ELEM=AMAX1(0.2*DWDY1,SIGH1)	CAVL0194
	IF (IAD.NE.0) ISAC=1	CAVL0195
	IF (ISAC.EQ.1) ELEM=.1*ELEM	CAVL0196
	IF (ELEM.LT.EMIN) ELEM=EMIN	CAVL0197
C		CAVL0198
C	BRANCH IF ANGLE IS SMALL AND LINE SOURCE IS LONG.	CAVL0199
C		CAVL0200
	IF (IAD.NE.0) GO TO 4311	CAVL0201
	IF (ASNF.GE.CAN) GO TO 4312	CAVL0202
4311	IF (DL1.GT. (1.5*ELEM)) GO TO 55	CAVL0203
C		CAVL0204
C	COMPUTE TRAVEL TIME AND DISPERSION COEFFICIENT FOR PSEUDO	CAVL0205
C	VERTICAL DISTANCE	CAVL0206
C		CAVL0207
4312	IF (DWDY.EQ.DWDY1) GO TO 4112	CAVL0208
	THRH=DWDY/WS36	CAVL0209
	SIGH=SIGY(JSTAB,THRH)	CAVL0210
4212	CONTINUE	CAVL0211
	THPV=DWDZ/WS36	CAVL0212
	SIGV=SIGZ(JSTAB,THRV)	CAVL0213
C		CAVL0214
C	EXPRESSIONS TO BE USED IN APPROXIMATIONS	CAVL0215
C		CAVL0216
	DENH2=2.*SIGH**2	CAVL0217
	DENZ2=2.*SIGV**2	CAVL0218
	D=SIGH*SIGV	CAVL0219
C		CAVL0220
C	ARGUMENTS OF EXPONENTIAL FUNCTION FOR VERTICAL DISPERSION	CAVL0221
C	USING SMALL ANGLE APPROXIMATION	CAVL0222
C		CAVL0223
	ARGZ1=-(ZR-Z1)**2/DENZ2	CAVL0224
	APGZ2=-(ZR+Z1)**2/DENZ2	CAVL0225
	IF (LSAC.EQ.1) GO TO 446	CAVL0226
	GO TO 445	CAVL0227
4111	SIGH1=SIGF	CAVL0228
	GO TO 4211	CAVL0229
4112	SIGH=SIGH1	CAVL0230
	GO TO 4212	CAVL0231
4113	SIGH1=SIGFO	CAVL0232
	GO TO 4211	CAVL0233
C		CAVL0234
C	LARGE ANGLE CASE: ARGUMENTS OF ERROR FUNCTIONS FOR	CAVL0235
C	HORIZONTAL DISPERSION	CAVL0236
C		CAVL0237
445	CONTINUE	CAVL0238
	APG=CSTH**2*SNFI**2*SIGV**2+SNTH**2*SIGH**2	CAVL0239
	RARG=SQRT(ARG)	CAVL0240
	A=FAFG/(1.4142*D)	CAVL0241
	AL=ELN*A	CAVL0242
	ARG1=(YR-Y1)*CSTH*SNFI*SIGV**2	CAVL0243
	ARG21=(ZR-Z1)*SNTH*SIGH**2	CAVL0244
	ARG22=-(ZR+Z1)*SNTH*SIGH**2	CAVL0245
	PA1=-(ARG1+ARG21)/(1.4142*D*RARG)	CAVL0246
		CAVL0247



	BA2=-(ARG1+ARG22)/(1.4142*D*FARG)	CAVL0248
	C1=(YR-Y1)**2/DENH2-ARGZ1	CAVL0249
	C2=(YR-Y1)**2/DENH2-ARGZ2	CAVL0250
C		CAVL0251
C	ARGUMENTS OF EXPONENTIAL FUNCITONS FOR VERTICAL DISPERSION	CAVL0252
C	USING THE GENERAL EQUATION	CAVL0253
C		CAVL0254
	ARGZ1=BA1**2-C1	CAVL0255
	ARGZ2=BA2**2-C2	CAVL0256
446	IF (ARGZ1.LT.-10.) GOTO 2411	CAVL0257
	IF (ARGZ2.GE.-10.) GOTO 2412	CAVL0258
	FAC1=EXP(ARGZ1)	CAVL0259
	FAC2=0	CAVL0260
	GC TO 39	CAVL0261
2411	IF (ARGZ2.LT.-10) GO TO 500	CAVL0262
	FAC1=0	CAVL0263
	FAC2=EXP(ARGZ2)	CAVL0264
	GC TO 39	CAVL0265
2412	IF (DWD.GT.XZ) GO TO 100	CAVL0266
C		CAVL0267
C	DOWNWIND DISTANCE IS LESS THAN THE CRITICAL DISTANCE: ONLY	CAVL0268
C	SOURCE AND GROUND REFLECTION ARE CONSIDERED	CAVL0269
C		CAVL0270
	FAC1=EXP(ARGZ1)	CAVL0271
	FAC2=EXP(ARGZ2)	CAVL0272
39	CCNTINUE	CAVL0273
C		CAVL0274
C	FIND THE LINEAR DISTRIBUTION OF POLLUTION ON THE	CAVL0275
C	RUNWAY FOR LANDING AND TAKE-OFF	CAVL0276
C		CAVL0277
	XSI2=(X-XST)**2+(Y-YST)**2+(Z-ZST)**2	CAVL0278
	XSI=SQRT(XSI2)	CAVL0279
	IF (X*XND.LT.0.OR.Y*YND.LT.0.OR.Z*ZND.LT.0) XSI=-XSI	CAVL0280
	IF (IAD.NE.0) CALL QMOD(XSI,QL)	CAVL0281
	QTOT=QTOT+QL*SEGL	CAVL0282
C		CAVL0283
C	STORE LAST VALUE OF CBAR	CAVL0284
C		CAVL0285
	CBARC=CBAR	CAVL0286
	IF (LSAC.EQ.1) GO TO 50	CAVL0287
C		CAVL0288
C	GENERAL DISPERSION EQUATION	CAVL0289
C		CAVL0290
	FJ1=FAC1*DIFEFF(BA1,AL)	CAVL0291
	FJ2=FAC2*DIFEFF(BA2,AL)	CAVL0292
	CBAR=CBAR+0.35355*COEFF1*Q1*(FJ1+FJ2)/(A*D)	CAVL0293
499	IF (CBAR.EQ.C) GO TO 49	CAVL0294
	IF (ABS((CBAR-CBAR)/CBAR).LE..00010) GO TO 600	CAVL0295
49	CCNTINUE	CAVL0296
	IF (NSUB.GT.1.AND.DLRS.GT.(.01*DL)) GO TO 60	CAVL0297
	GO TO 600	CAVL0298
C		CAVL0299
C	SMALL-ANGLE APPFOXIMATION	CAVL0300
C		CAVL0301
50	ARGYY=-(YR-Y1)**2/DENH2	CAVL0302
	IF (ARGYY.LT.-10.) GO TO 500	CAVL0303
	FAC=0.5*(FAC1+FAC2)	CAVL0304
	BRAC=EXP(ARGYY)	CAVL0305
	CBAR=CBAR+COEF2*QL*DLN*FAC*BRAC/D	CAVL0306
	GO TO 499	CAVL0307
C		CAVL0308
C	ANGLE IS SMALL AND SOURCE IS LONG	CAVL0309

C		CAVL0310
	55 ISUF=1	CAVL0311
	XRS=X2-X1	CAVL0312
	YPS=Y2-Y1	CAVL0313
	ZRS=Z2-Z1	CAVL0314
	X2=X1	CAVL0315
	Y2=Y1	CAVL0316
	Z2=Z1	CAVL0317
C		CAVL0318
C	COMPUTE COORDINATES FOR NEXT LINE SEGMENT	CAVL0319
C		CAVL0320
	60 NSUB=1.+DIRS/FLEM	CAVL0321
	RSUB=NSUB	CAVL0322
	SEGL=DIRS/RSUB	CAVL0323
	DELX=XPS/RSUB	CAVL0324
	DELY=YRS/RSUB	CAVL0325
	DELZ=ZRS/RSUB	CAVL0326
	X1=X2	CAVL0327
	Y1=Y2	CAVL0328
	Z1=Z2	CAVL0329
	X2=X2+DELX	CAVL0330
	Y2=Y2+DELY	CAVL0331
	Z2=Z2+DELZ	CAVL0332
	DLRS=DIRS-SEGL	CAVL0333
	XFS=XPS-DELX	CAVL0334
	YRS=YRS-DELY	CAVL0335
	ZRS=ZRS-DELZ	CAVL0336
	X=.5*(X1+X2)	CAVL0337
	Y=.5*(Y1+Y2)	CAVL0338
	Z=.5*(Z1+Z2)	CAVL0339
C		CAVL0340
C	GO BACK TO COMPUTE CONTRIBUTION FROM NEXT SEGMENT	CAVL0341
C		CAVL0342
	GC TC 5	CAVL0343
C		CAVL0344
C	DOWNWIND DISTANCE IS GREATER THAN, BUT LESS THAN TWICE, THE	CAVL0345
C	CRITICAL DISTANCE. LINEAR INTERPOLATION IS USED	CAVL0346
C		CAVL0347
	100 YL = Y1	CAVL0348
	Z1 = Z1	CAVL0349
	IF (Z1 .LE. Z2) GO TO 105	CAVL0350
	YL = Y2	CAVL0351
	Z1 = Z2	CAVL0352
	105 FAC3=0.5*SIGV/(COEF1*HLIDM)	CAVL0353
	IF (LWD.GT.2.*XZ) GO TO 200	CAVL0354
	DENZ1=0.47*HLIDM	CAVL0355
	DENZ12=DENZ1**2	CAVL0356
	IF (ISAC.EQ.1) GO TO 101	CAVL0357
	102 TL=XZ/WS36	CAVL0358
	DENHL2=2.*SIGY(JSTAB,TL)**2	CAVL0359
	DENO=CSTH**2*SNFI**2*DENZ12 +SNTH**2*DENHL2	CAVL0360
	ARGZ1=-((YR-YL)*SNTH-(ZR-ZL)*CSTH*SNFI)**2/DENO	CAVL0361
	ARGZ2=-((YR-YL)*SNTH-(ZR+ZL)*CSTH*SNFI)**2/DENO	CAVL0362
	GC TC 103	CAVL0363
	101 ARGZ1=- (ZR-ZL)**2/DENZ12	CAVL0364
	ARGZ2=- (ZF+ZL)**2/DENZ12	CAVL0365
	103 FAC1=EXP(ARGZ1)	CAVL0366
	FAC2=EXP(ARGZ2)	CAVL0367
	FAC1=FAC1+(DWD-XZ)*(FAC3-FAC1)/XZ	CAVL0368
	FAC2=FAC2+(DWD-XZ)*(FAC3-FAC2)/XZ	CAVL0369
	GC TC 39	CAVL0370
C		CAVL0371

C DOWNWIND DISTANCE IS BEYOND 2 TIMES THE CRITICAL DISTANCE,  
C UNIFORM MIXING IS ASSUMED

C

200 FAC1=FAC3

FAC2=FAC3

GO TO 39

500 IF(DLRS.LT. (.01\*DL)) GO TO 600

IF(ISAC.EQ.1.AND.NSUB.EQ.0) GO TO 55

IF(NSUB.GE.1) GO TO 60

600 IF (NSUB.NE.1.OR.XMAX.GE.XP) QTOT=1.0

IF (QTOT.EQ.0.0) QTOT=1.0

C

C

NORMALIZE CBAR TO THE TOTAL POLLUTANT DENSITY CALCULATED

C

ALONG THE LINE

C

CAVL=CBAR/QTOT

RETURN

END

CAVL0372

CAVL0373

CAVL0374

CAVL0375

CAVL0376

CAVL0377

CAVL0378

CAVL0379

CAVL0380

CAVL0381

CAVL0382

CAVL0383

CAVL0384

CAVL0385

CAVL0386

CAVL0387

CAVL0388

CAVL0389



## SUBROUTINE CLASSE

### Purpose:

To print an error message if the wrong ICLASS value is input to one of the airbase non-aircraft or environ emission distribution subroutines.

### Input:

None

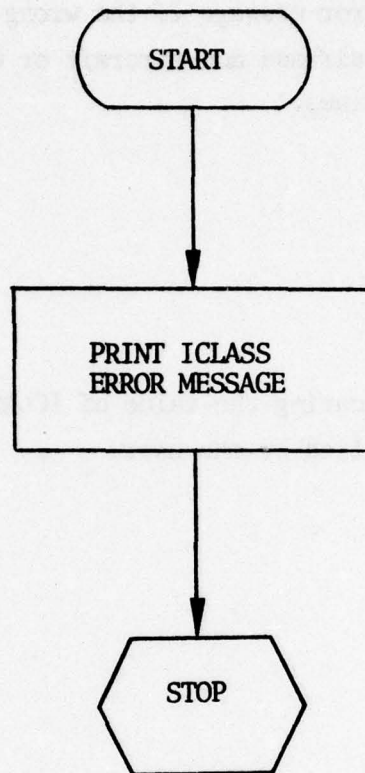
### Output:

A message indicating the value of ICLASS set by the code and the value supplied by the user.

### Subroutines Called:

None

SUBROUTINE CLASSE



```

C      SUBFCUTINE CLASSE  (I,J)
C
C      THIS FCUTINE PRINTS THE ICLASS ERROR MESSAGE
C
      PRINT 1, I,J
1  FORMAT(17H0ICLASS SHOULD BE,I4,18H, INPUT CARD READS,I4)
      STOP
      END

```

```

CLASE000
CLASE001
CLASE002
CLASE003
CLASE004
CLASE005
CLASE006
CLASE007

```



## SUBROUTINE DEPART

### Purpose:

To calculate the points in the runway roll and climbout modes as a function of aircraft type using current meteorological conditions and airbase specific pressure altitudes and airbase dependent basic aircraft parameters.

### Input:

Basic aircraft data, current meteorological conditions, runway data, aircraft identification.

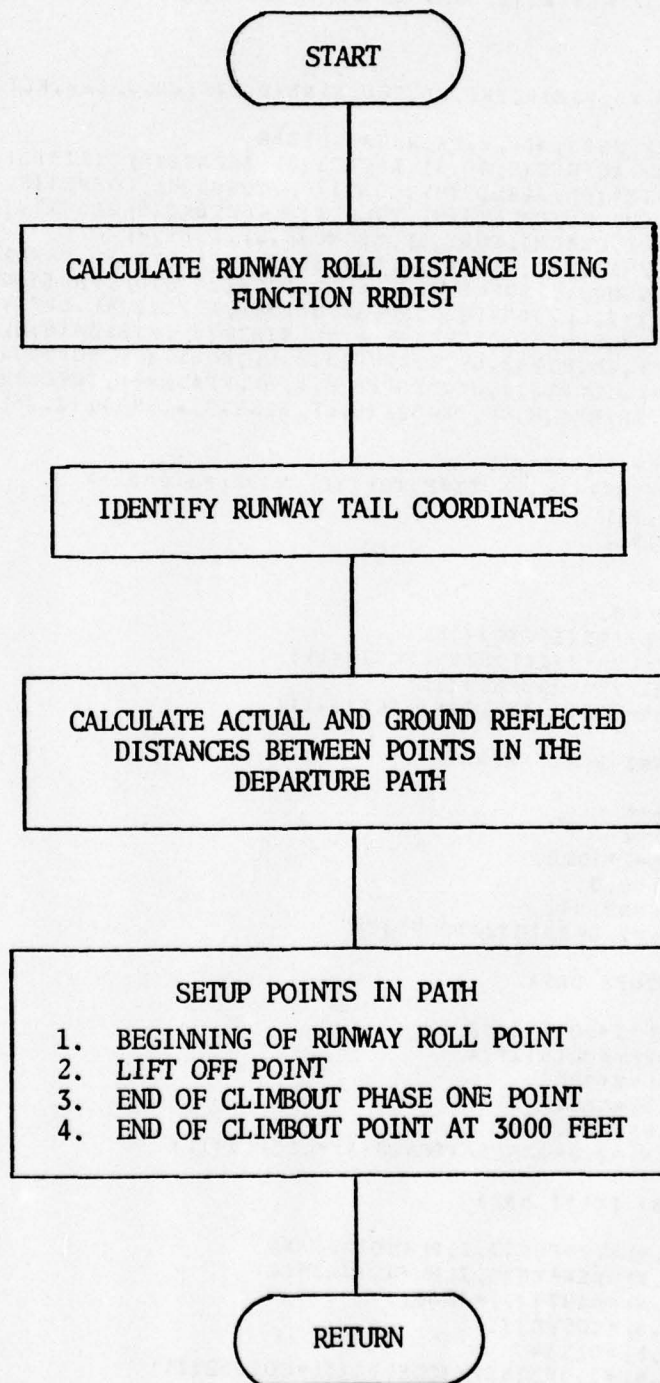
### Output:

Points in departure path as a function of runway and aircraft type.

### Subroutine Called:

Function RRDIST.

# SUBROUTINE DEPART



C	SUBROUTINE DEPART(N,I)	DEPRT000
C	THIS ROUTINE CALCULATES THE POINTS IN THE DEPARTURE PATH	DEPRT001
C	AS A FUNCTION OF RUNWAY(N) AND AIRCRAFT TYPE(I)	DEPRT002
C		DEPRT003
	REAL INDSPD	DEPRT004
	INTEGER ENGNO	DEPRT005
	COMMON / MET / WS,WSMEH,IWS,WD,IWD,SINEWD,CVSEWD,JSTAB,HLID,TEMP,	DEPRT006
	TEMP	DEPRT007
	CCHMCN /ANNMET/ TBAR,ACD,P,PA,WSBAR,DTBAR	DEPRT008
	COMMON /ACEDB1/ ACMEFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),	DEPRT009
	. INDSFD(8),APSPD1(8),AFSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),	DEPRT010
	. COSPD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),	DEPRT011
	. APFHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)	DEPRT012
	COMMON /ACEDB2/ NACTYP,NRWYS,NPKAR,IEGFLG,IACTYP(8),ANNAPR(8),	DEPRT013
	. ANNDEP(8),ANNTOGO(8),ARRFCN(24,8,6),DEPFCN(24,8,6),TGO(3,4,8),	DEPRT014
	. DISFNW(6),RWNY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),	DEPRT015
	. ACSPII(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),	DEPRT016
	. IIBSEG(16,8,6),IDIBTW(8,6),TTARPR(8,8,6),NOBTI(6),NOBSEG(8,6),	DEPRT017
	. IOBSEG(16,8,6),IDOBTW(8,6),TTDPFR(8,8,6),NPASQ(6),IDPRKA(6),	DEPRT018
	. PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JFS1(8)	DEPRT019
	RD=RWNY(7,N)	DEPRT020
	WSPD=WS*1.9426*COS(WD-RD)	DEPRT021
	HDIS12=RRDIST(IDRR(I),PA,TEMP,TOWT(I),WSPD)*3.048E-4	DEPRT022
	XA=SIN(RWNY(7,N))	DEPRT023
	YA=COS(RWNY(7,N))	DEPRT024
	X=RWNY(2,N)	DEPRT025
	Y=RWNY(3,N)	DEPRT026
	Z=RWNY(4,N)/1000.	DEPRT027
	DIS23=COHT1(I)/SIN(ASCNT1(I))	DEPRT028
	DIS34=(CLMBHT-COHT1(I))/SIN(ASCNT2(I))	DEPRT029
	HDIS23=COHT1(I)/TAN(ASCNT1(I))	DEPRT030
	HDIS34=(CLMBHT-COHT1(I))/TAN(ASCNT2(I))	DEPRT031
C		DEPRT032
C	START OF RUNWAY ROLL DATA	DEPRT033
C		DEPRT034
	DEPFCN(1,I,N)=X	DEPRT035
	DEPFCN(2,I,N)=Y	DEPRT036
	DEPFCN(3,I,N)=Z*1000.	DEPRT037
	DEPFCN(4,I,N)=0.0	DEPRT038
	DEPFCN(5,I,N)=HDIS12	DEPRT039
	DEPFCN(6,I,N)=2.0*HDIS12/TOSPD(I)	DEPRT040
C		DEPRT041
C	FCINT CF LIFTOFF DATA	DEPRT042
C		DEPRT043
	DEPFCN(7,I,N)=X+HDIS12*XA	DEPRT044
	DEPFCN(8,I,N)=Y+HDIS12*YA	DEPRT045
	DEPFCN(9,I,N)=Z*1000.	DEPRT046
	DEPFCN(10,I,N)=TOSPD(I)	DEPRT047
	DEPFCN(11,I,N)=DIS23	DEPRT048
	DEPFCN(12,I,N)=2.0*DIS23/(TOSPD(I)+COSPD1(I))	DEPRT049
C		DEPRT050
C	END OF CLIMB1 FCINT DATA	DEPRT051
C		DEPRT052
	DEPFCN(13,I,N)=DEPFCN(7,I,N)+HDIS23*XA	DEPRT053
	DEPFCN(14,I,N)=DEPFCN(8,I,N)+HDIS23*YA	DEPRT054
	DEPFCN(15,I,N)=COHT1(I)*1000.	DEPRT055
	DEPFCN(16,I,N)=COSPD1(I)	DEPRT056
	DEPFCN(17,I,N)=DIS34	DEPRT057
	DEPFCN(18,I,N)=2.0*DIS34/(COSPD1(I)+COSPD2(I))	DEPRT058
C		DEPRT059
C	END OF CLIMBOUT POINT DATA	DEPRT060
		DEPRT061



C

DEPFCN (19,I,N)=DEPFCN (13,I,N)+HDIS34\*YA  
DEPFCN (20,I,N)=DEPFCN (14,I,N)+HDIS34\*YA  
DEPFCN (21,I,N)=CLMBHT\*1000.  
DEPFCN (22,I,N)=COSPD2 (I)  
RETUEN  
END

DEPRT062  
DEPRT063  
DEPRT064  
DEPRT065  
DEPRT066  
DEPRT067  
DEPRT068

# FUNCTION DIFERF(X,PH)

## Purpose:

To find the difference between two error functions,  $\text{erf}(X+PH) - \text{erf}(X)$ .

## Input:

X and PH

## Output:

The difference between the error functions

## Procedure:

1. If  $PH \leq .05$ , the formula given in the Handbook of Mathematical Functions, National Bureau of Standards, Applied Mathematics Series 55 is used:

$$\text{DIFERF} = 1.12838 \cdot PH \cdot e^{-X^2} [1 - PH \cdot X + (2 \cdot X^2 - 1) \cdot PH^{2/3}]$$

2. If  $PH > .05$  and X and X+PH are of different sign:

$$\text{DIFERF} = \text{erf}(X+PH) - \text{erf}(X)$$

3. If  $PH > .05$  and X and X+PH are both negative:

$$\text{DIFERF} = -1. * [\text{erfc}(-X) - \text{erfc}(-X-PH)]$$

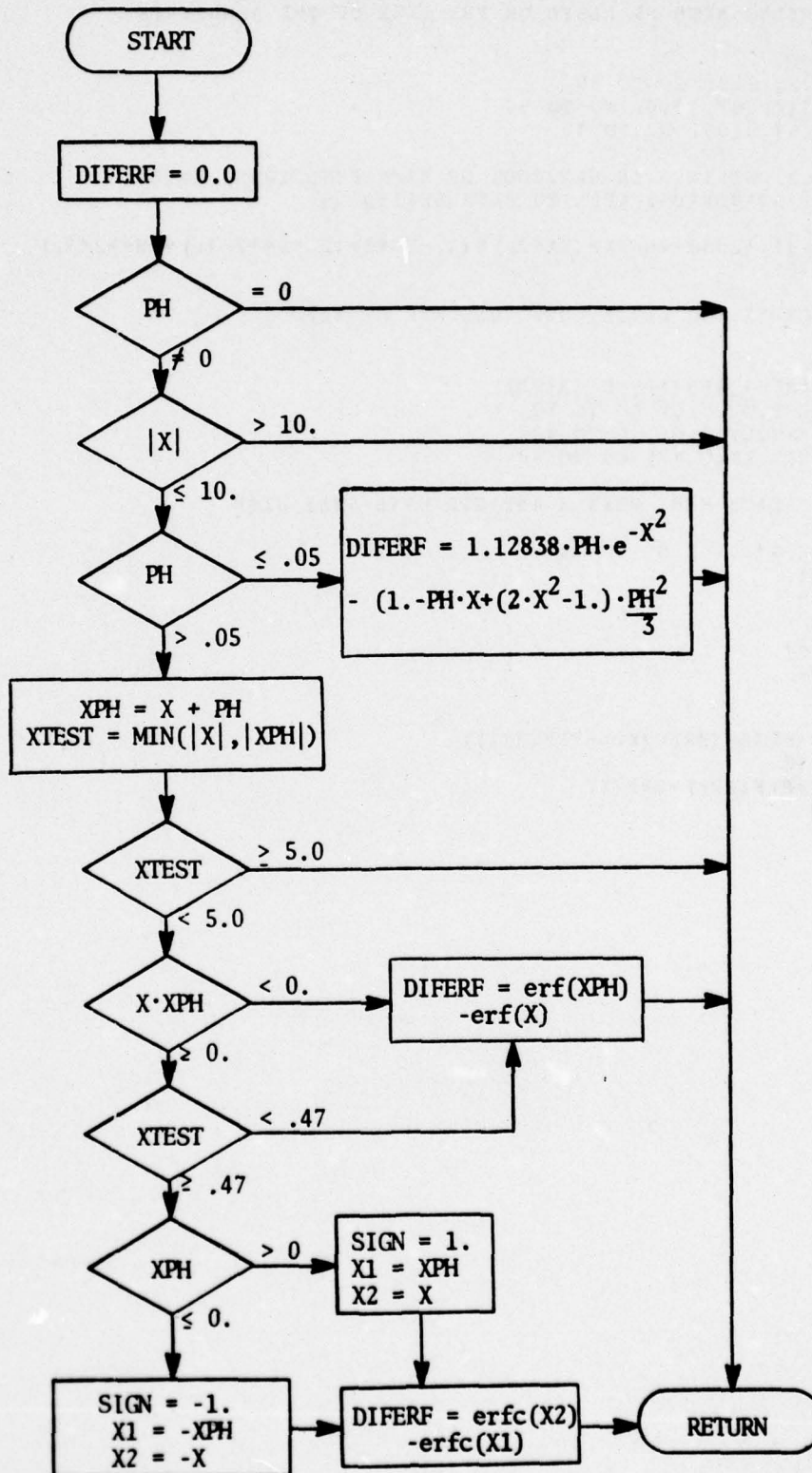
4. If  $PH > .05$  and X and X+PH are both positive:

$$\text{DIFERF} = \text{erfc}(X) - \text{erfc}(X+PH)$$

## Function Called:

ERF, ERFC

FUNCTION DIFERF(X,PH)





C	FUNCTION DIFERF(X,PH)	DIFER000
C	THIS FUNCTION FINDS THE DIFFERENCE BETWEEN TWO ERROR FUNCTIONS	DIFER001
C	USING VARYING METHODS BASED ON THE SIZE OF THE ARGUMENTS	DIFER002
C		DIFER003
	DIFERF=0.	DIFER004
	IF (PH.EQ.0.0) GO TO 50	DIFER005
	IF (ABS(X).GT.10.0) GO TO 50	DIFER006
	IF (PH.GT.0.05) GO TO 10	DIFER007
C		DIFER008
C	USE METHOD OUTLINED IN HANDBOOK OF MATH FUNCTIONS, NATL	DIFER009
C	BUREAU OF STANDARDS, APPLIED MATH SERIES 55	DIFER010
C		DIFER011
	DIFERF=(1.12838*PH/EXP(X**2))*(1.-PH*X+(2.*X**2-1.)*PH**2/3.)	DIFER012
	GO TO 50	DIFER013
C		DIFER014
C	DIFFERENCE IS TOO LARGE, MUST USE ERF OR ERFC	DIFER015
C		DIFER016
	10 XPH=X+PH	DIFER017
	XTEST=AMIN1(ABS(X),ABS(X+PH))	DIFER018
	IF (XTEST.GE.5.0) GO TO 50	DIFER019
	IF (X*XPH.LT.0.0) GO TO 40	DIFER020
	IF (XTEST.LT.0.47) GO TO 40	DIFER021
C		DIFER022
C	CAN ONLY REACH HERE WHEN X AND XPH HAVE SAME SIGN	DIFER023
C		DIFER024
	IF (XPH.GT.0.0) GO TO 20	DIFER025
	SIGN=-1.	DIFER026
	X1=-XPH	DIFER027
	X2=-X	DIFER028
	GO TO 30	DIFER029
	20 SIGN=1.	DIFER030
	X1=XPH	DIFER031
	X2=X	DIFER032
	30 DIFERF=SIGN*(ERFC(X2)-ERFC(X1))	DIFER033
	GO TO 50	DIFER034
	40 DIFERF=ERF(XPH)-ERF(X)	DIFER035
	50 RETURN	DIFER036
	END	DIFER037
		DIFER038

## SUBROUTINE EMISAR

### Purpose:

To add emissions from a given activity to all others contained in the specified geometric area or line.

### Input:

None

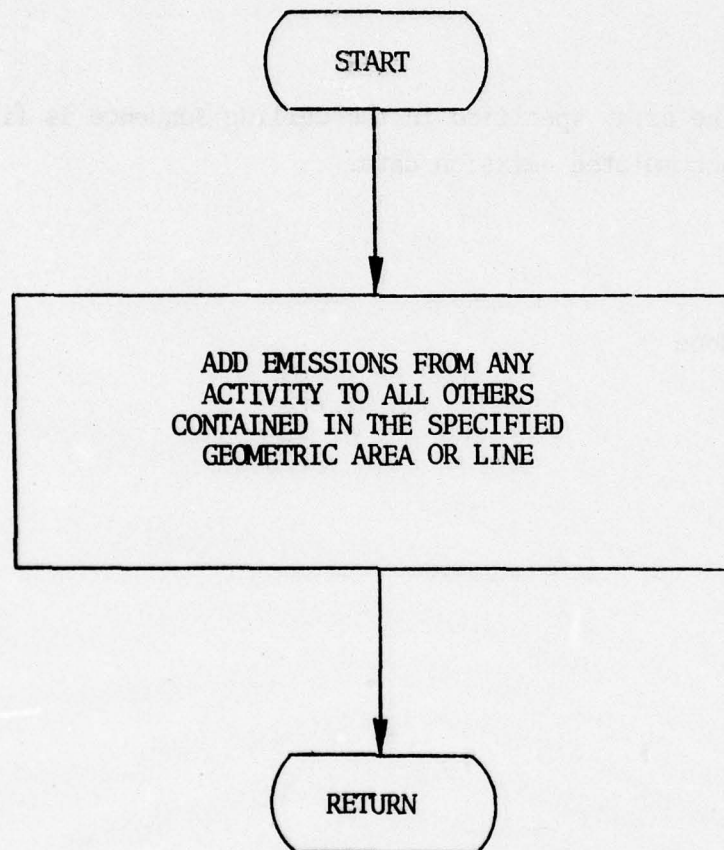
### Output:

The array specified in the calling sequence is filled with the accumulated emission data.

### Subroutines Called:

None

SUBROUTINE EMISAR





C	SUBROUTINE EMISAR(MAXN,ARRAY,I1,I2)	EMISR000
C	THIS ROUTINE ACCUMULATES EMISSIONS FROM ANY ACTIVITY WITH	EMISR001
C	OTHERS CONTAINED IN THE SAME AIRBASE AREA OR LINE.	EMISR002
C	MAXN = NO. OF SOURCES IN AN ACTIVITY	EMISR003
C	ARRAY = SPECIFIED AREA OR LINE OUTPUT ARRAY	EMISR004
C	I1,I2 = DIMENSIONS OF ARRAY	EMISR005
C	NSRCE = POINTER TO LOCATION OF SOURCES IN SOURCE	EMISR006
C	LCC1 = POINTER TO LOCATION OF LIST OF EMISSIONS IN ARRAY	EMISR007
C	SOURCE(2,N) = POINTER TO LOCATION OF SOURCE AREA OR LINE	EMISR008
C		EMISR009
	COMMON /SRCE, NPLTS, NENPT, NENAR, NENLN, NABPT, NABAR, NABLN,	EMISR010
	. NACET, NACAR, NACLN, ENET(16,100), ENAR(11,100), ENLN(14,20),	EMISR011
	. ABET(16,150), ABAR(11,100), ABLN(14,100)	EMISR012
	COMMON/JUNK/DAYS,LSRCE,NSRCE,SOURCE(17,300),SORGM(10,200)	EMISR013
	. ,LCC1,LCC2,NGEOM,IPT	EMISR014
	DIMENSION ARRAY(I1,I2)	EMISR015
	LSRCE=NSRCE+1	EMISR016
	NSRCE=NSRCE+MAXN	EMISR017
C		EMISR018
	DC 10 N=LSRCE,NSRCE	EMISR019
	J=SOURCE(2,N)	EMISR020
	DC 10 I=1,NPLTS	EMISR021
	ARRAY(I+LOC1,J)=ARRAY(I+LCC1,J)+SOURCE(I+2,N)	EMISR022
10	CONTINUE	EMISR023
	RETURN	EMISR024
	END	EMISR025
		EMISR026

## SUBROUTINE ENARAY

### Purpose:

1. To read from the master source tape all data needed to define environ point, area and line sources.
2. To compute the emission rates due to point sources, stationary, mobile, land use or combined area sources and roadway and non-roadway line sources.

### Input:

If the diurnal distribution cards are input, an additional parameter, IMETH, is input here to choose the method of distribution of emissions from those land use or combined area source activities not using the default of a uniform distribution.

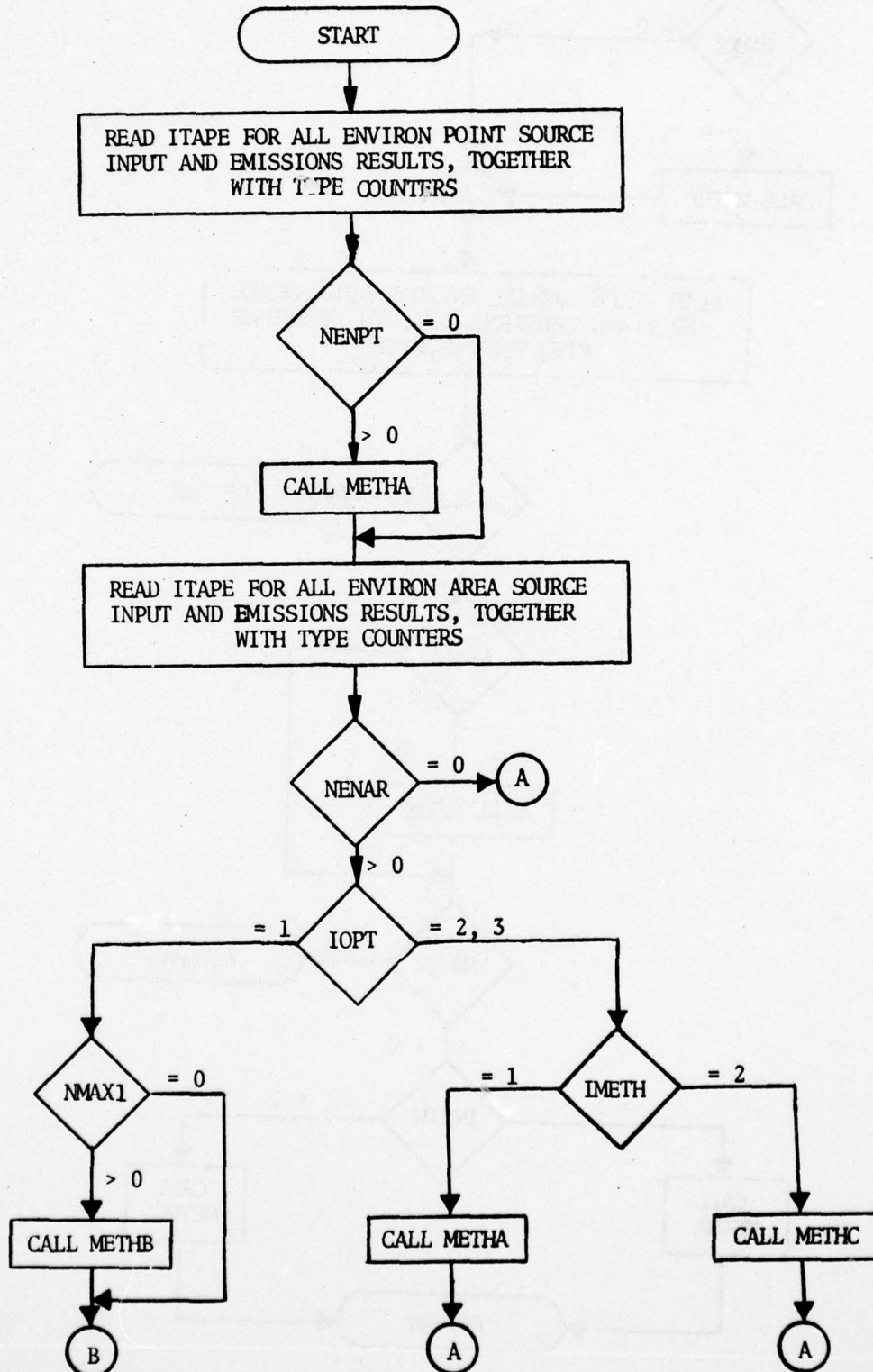
### Output:

The arrays, ENPT, ENAR, and ENLN, are filled with geometry and emission data for all environ sources.

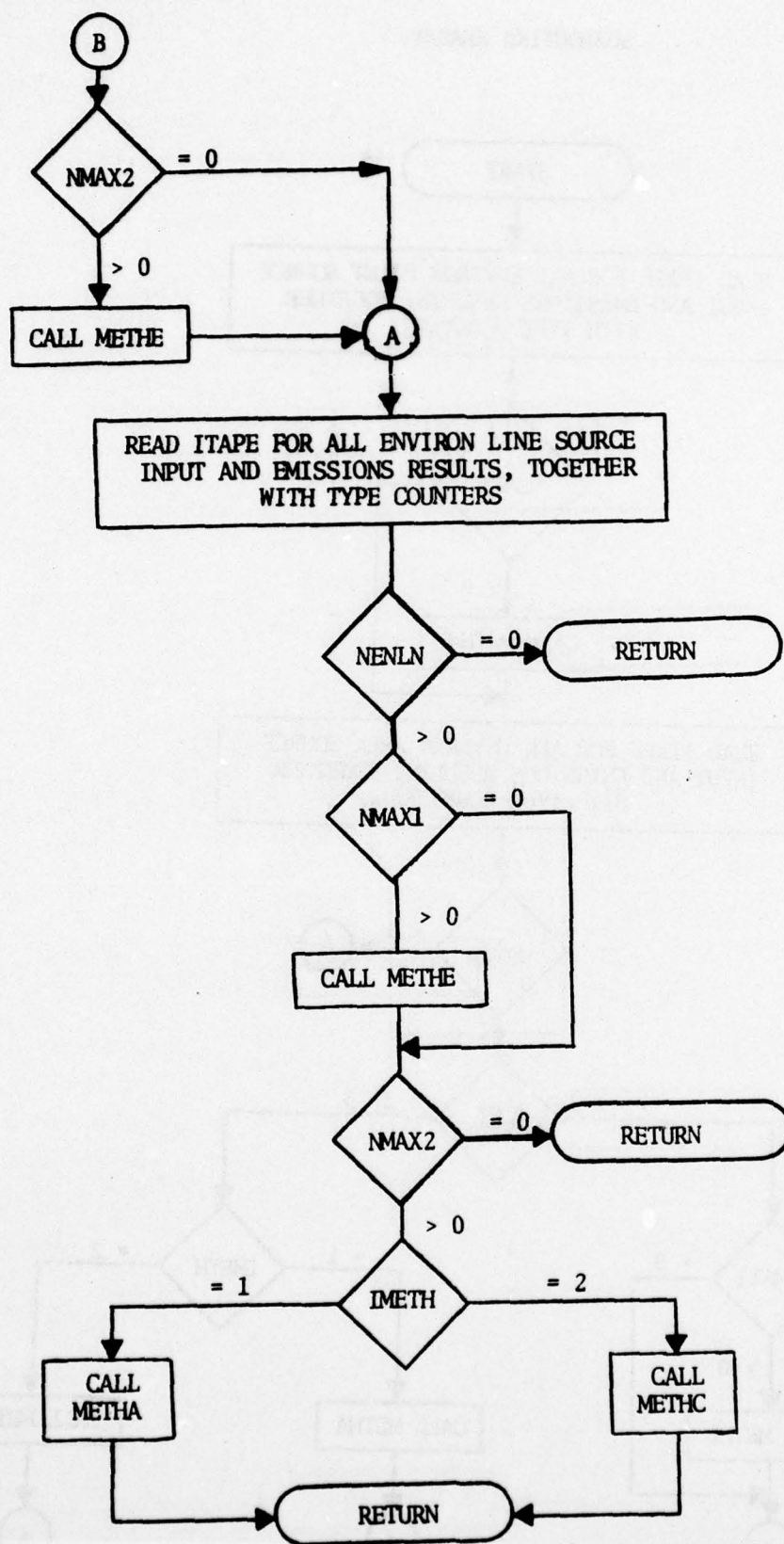
### Subroutines Called:

METHA, METHB, METHC, METHD

# SUBROUTINE ENARAY







C	SUBFCUTINE ENARAY	ENARY000
C	THIS ROUTINE COMPUTES THE EMISSION RATES FOR ALL	ENARY001
C	ENVIFCN SOURCES	ENARY002
C		ENARY003
	CCMMON / DEFAULT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	ENARY004
	CCMMCN/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)	ENARY005
	. ,LOC1,LOC2,NGEOM,IPT	ENARY006
	CCMMON/MONMET/TMBAR,WSMBAF,AMDMBR,DTMBAR	ENARY007
	COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,	ENARY008
	. JSTAE,HLID,TEMP,TEMK	ENARY009
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG	ENARY010
	CCMMCN /DSTRBT/ ACMO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),	ENARY011
	. VHMLDY(2),VHMLHR(24),CVAABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),	ENARY012
	. CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1	ENARY013
	CCMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAE,NABLN,	ENARY014
	. NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),	ENARY015
	. ABPT(16,150),ABAR(11,100),ABLN(14,100)	ENARY016
C		ENARY017
C	C****POINTS	ENARY018
C		ENARY019
	READ(ITAPE) NFNPT,NTOT,((SORCE(I,N),I=1,NTOT),N=1,NENPT)	ENARY020
	IF (NENPT.EQ.0) GO TO 100	ENARY021
	ICLASS=201	ENARY022
	LCC1=10	ENARY023
	LCC2=11	ENARY024
	NGEOM=9	ENARY025
	I1=16	ENARY026
	I2=100	ENARY027
	IFT=1	ENARY028
	NSRCE=0	ENARY029
	CALL METHA(NENPT,ENPT,I1,I2,ICLASS)	ENARY030
C		ENARY031
C	C****AREAS	ENARY032
C		ENARY033
	100 READ(ITAPE) NENAR,NTOT,IOPT,NMAX1,NMAX2,	ENARY034
	1 ((SORCE(I,N),I=1,NTOT),N=1,NENAR)	ENARY035
	IF (NENAR.EQ.0) GO TO 300	ENARY036
	LCC1=5	ENARY037
	LCC2=7	ENARY038
	NGECM=5	ENARY039
	IFT=0	ENARY040
	I1=11	ENARY041
	I2=100	ENARY042
	NSRCE=0	ENARY043
	GC TC (110,120,130),IOPT	ENARY044
C		ENARY045
C	C****CPTION 1 STATIONARY AREAS	ENARY046
C		ENARY047
	110 ICLASS=202	ENARY048
	IF (NMAX1.GT.0)	ENARY049
	1 CALL METHB(NMAX1,ENAR,I1,I2,ICLASS)	ENARY050
C		ENARY051
C	C****OPTION 1 MOBILE AREAS	ENARY052
C		ENARY053
	IF (NMAX2.GT.0)	ENARY054
	1 CALL METHC(NMAX2,ENAR,CVENMO,CVENDY,CVENHR,I1,I2)	ENARY055
	GC TC 300	ENARY056
C		ENARY057
C	C****CPTION 2 OF 3 LAND USE OR COMBINED AREAS	ENARY058
C		ENARY059
	120 ICLASS=203	ENARY060
		ENARY061

GC TC 200	ENARY062
130 ICLASS=204	ENARY063
200 IMETH=1	ENARY064
IF (JFLAG.EQ.0) READ 201,IMETH	ENARY065
201 FCRMAT(I4)	ENARY066
IF (IMETH.EQ.1) CALL METHA(NMAX1,ENAR,I1,I2,ICLASS)	ENARY067
IF (IMETH.EQ.2) CALL METHC(NMAX1,ENAR,I1,I2,ICLASS)	ENARY068
C	ENAPY069
C****LINES	ENARY070
C NMAX1 = NO. OF ROADWAY LINES	ENARY071
C NMAX2 = NO. OF NON-ROADWAY LINES	ENARY072
C	ENARY073
300 READ(ITAPE) NENLN,NTOT,NMAX1,NMAX2,	ENARY074
1 ((SORCE(I,N),I=1,NTOT),N=1,NENLN)	ENARY075
IF (NENLN.EQ.0) GO TO 400	ENARY076
LCC1=8	ENARY077
LCC2=10	ENARY078
NGEOM=8	ENARYC79
NSRCE=0	ENARY080
I1=14	ENARY081
I2=20	ENARY082
IFT=0	ENARY083
IF (NMAX1.GT.C)	ENARY084
1 CALL METHE(NMAX1,ENLN,CVENMO,CVENDY,CVENHR,I1,I2)	ENARY085
C	ENARY086
IF (NMAX2.EQ.C) GO TO 400	ENARY087
ICLASS=206	ENARY088
IMETH=1	ENARY089
IF (JFLAG.EQ.0) READ 201,IMETH	ENARY090
IF (IMETH.EQ.1) CALL METHA(NMAX2,ENLN,I1,I2,ICLASS)	ENARY091
IF (IMETH.EQ.2) CALL METHC(NMAX2,ENLN,I1,I2,ICLASS)	ENARY092
400 RETURN	ENARY093
END	ENARY094



SUBROUTINE INDINP  
ENTRY DEPINP

Purpose:

To print the input parameters for both wind independent and wind dependent sources.

Input:

All source parameters.

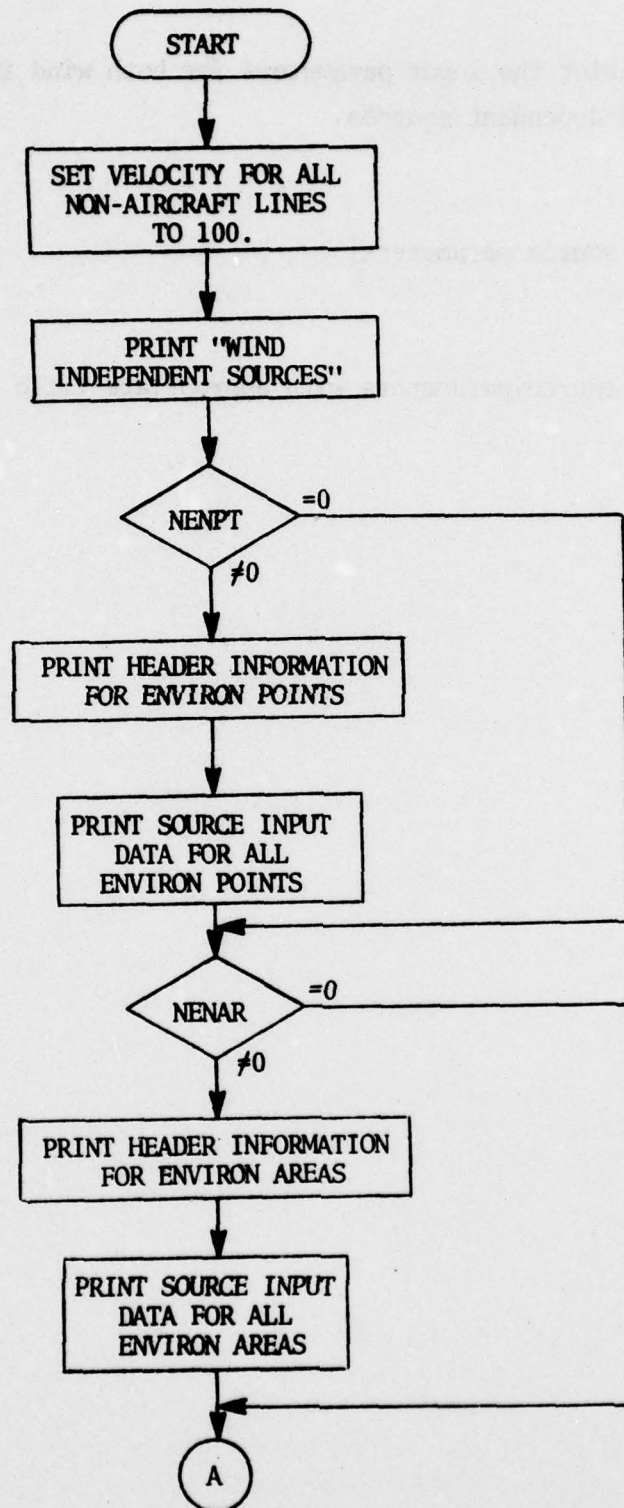
Output:

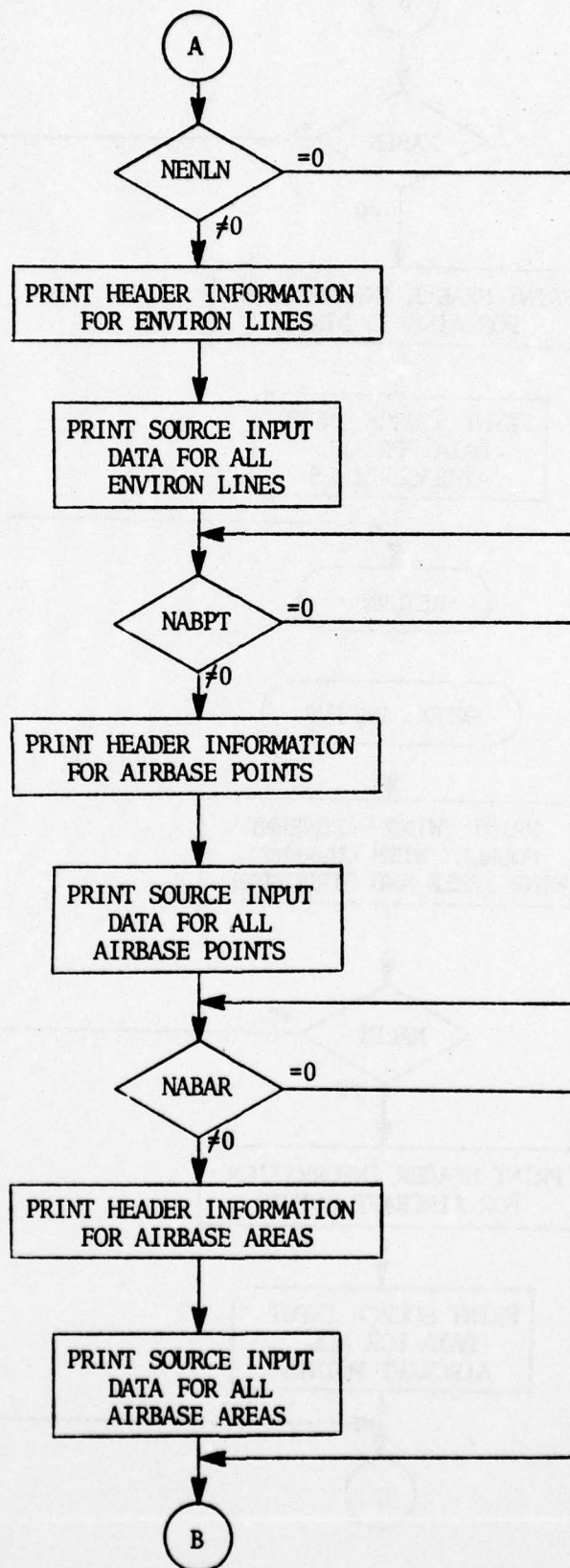
All source parameters with appropriate title information.

Subroutines  
Called:

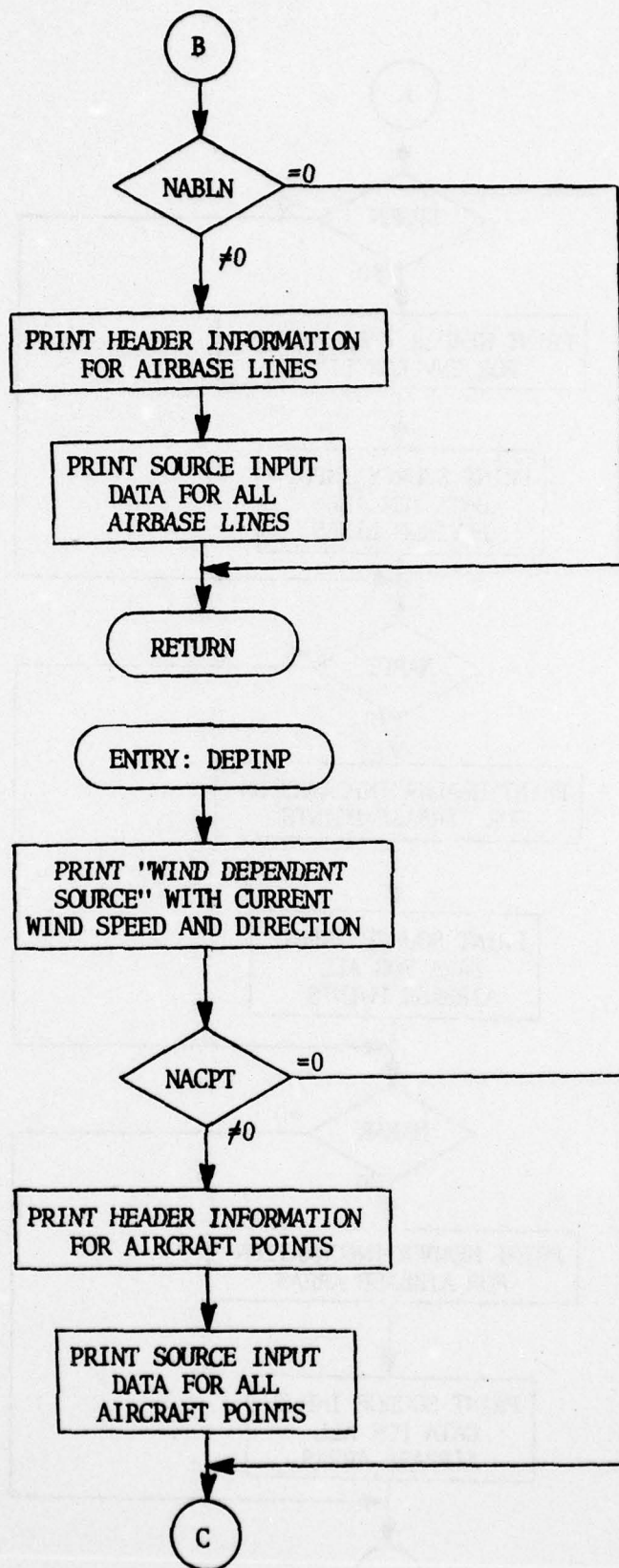
None

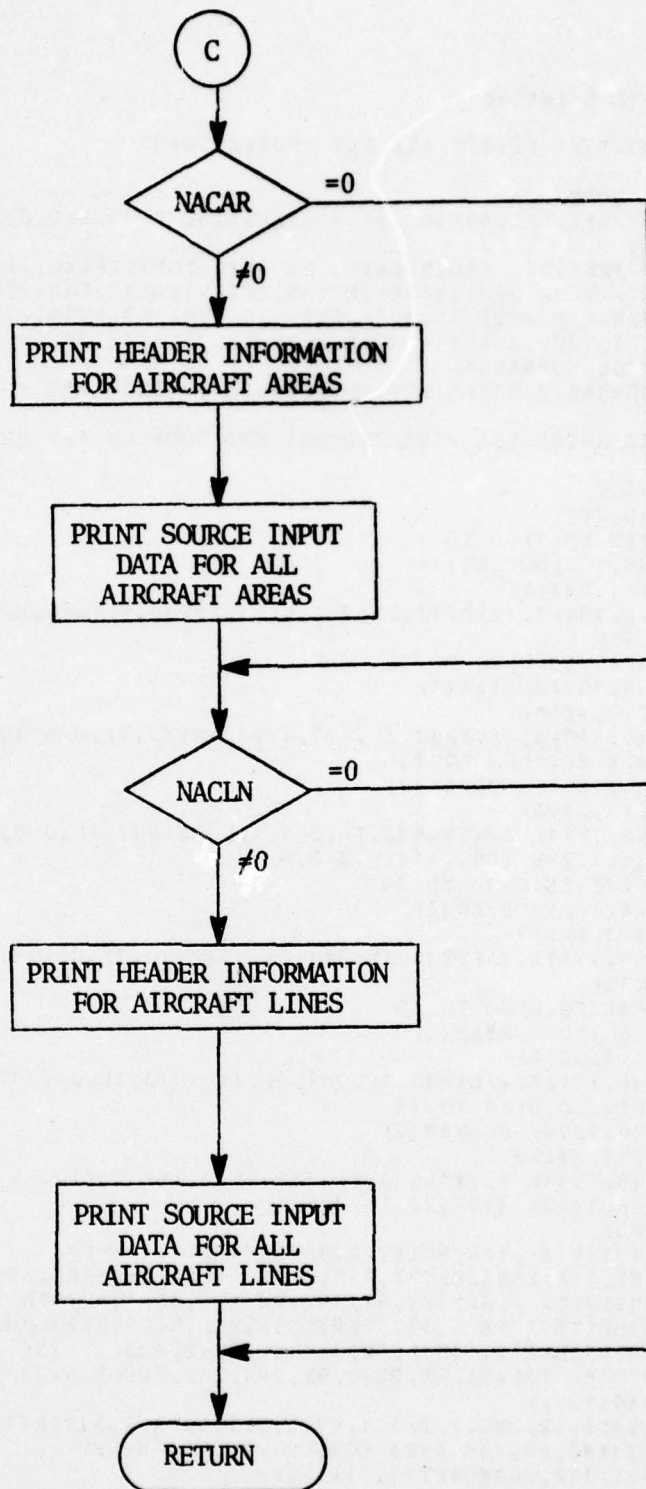
SUBROUTINE INDINP  
(ENTRY: DEPINP)











	SUBROUTINE INDINP	INDIP000
C		INDIP001
C	THIS ROUTINE PRINTS ALL THE SOURCE INPUT	INDIP002
C		INDIP003
	REAL*8 SORNAM	INDIP004
	CCMMCN /MET/ WS,WSMFH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP	INDIP005
	.,TEMK	INDIP006
	COMMON /PERIOD/ IMO,NCDAYS,IDY,IHR1,IHR2,IFLAG,JFLAG	INDIP007
	COMMON /SRCE/ NFOL,NENFT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,	INDIP008
	. NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),	INDIP009
	. ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)	INDIP010
	DIMENSION SORNAM(3)	INDIP011
	DATA SCRNAM / 8HENVIRON ,8HAIRBASE ,8HAIRCRAFT /	INDIP012
C		INDIP013
C	AT THIS ENTRY ALL WIND INDEPENDENT SOURCES ARE PRINTED	INDIP014
C		INDIP015
	ENLNA=1.0	INDIP016
	WRITE(6,200)	INDIP017
	IF(NENFT.EQ.0)GO TO 11	INDIP018
	WRITE(6,100) SORNAM(1)	INDIP019
	DC 1 I=1,NENFT	INDIP020
	1 WRITE(6,101) I, (ENPT(J,I),J=1,4), (ENPT(J,I),J=6,8), (ENFT(J,I),	INDIP021
	. J=10,15)	INDIP022
	11 IF(NENAR.EQ.0)GO TO 12	INDIP023
	WRITE(6,110) SORNAM(1)	INDIP024
	DC 2 I=1,NENAR	INDIP025
	2 WRITE(6,111) I, (ENAR(J,I),J=1,4), (ENAR(J,I),J=6,10)	INDIP026
	12 IF(NENLN.EQ.0)GO TO 13	INDIP027
	WRITE(6,1200) SORNAM(1)	INDIP028
	DO 5 I=1,NENLN	INDIP029
	WRITE(6,1211) I, (ENLN(J,I),J=1,4), (ENLN(J,I),J=9,13)	INDIP030
	5 WRITE(6,1222) (ENLN(J,I),J=6,8)	INDIP031
	13 IF(NABPT.EQ.0)GO TO 14	INDIP032
	WRITE(6,100) SORNAM(2)	INDIP033
	DO 3 I=1,NABPT	INDIP034
	3 WRITE(6,101) I, (ABPT(J,I),J=1,4), (ABPT(J,I),J=6,8), (ABPT(J,I),	INDIP035
	. J=10,15)	INDIP036
	14 IF(NABAR.EQ.0)GO TO 15	INDIP037
	WRITE(6,110) SCRNAM(2)	INDIP038
	DO 4 I=1,NABAR	INDIP039
	4 WRITE(6,111) I, (ABAR(J,I),J=1,4), (ABAR(J,I),J=6,10)	INDIP040
	15 IF(NABLN.EQ.0)GO TO 16	INDIP041
	WRITE(6,1200) SCRNAM(2)	INDIP042
	DC 6 I=1,NABLN	INDIP043
	WRITE(6,1211) I, (ABLN(J,I),J=1,4), (ABLN(J,I),J=9,13)	INDIP044
	6 WRITE(6,1222) (ABLN(J,I),J=6,8)	INDIP045
	16 CONTINUE	INDIP046
	100 FORMAT(1H0,A8,14H POINT SOURCES/1X,119(1H-)/	INDIP047
	. 8X,1HI,11X,8HGEOMETRY,11X,22HI STACK PARAMETERS I,4X,1HI/	INDIP048
	. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,12HWIDTH I TEMP,4X,3HVEL,	INDIP049
	. 3X,11HDIAM I PR I,13X,28HEMISSIONS(MICROGRAMS/SECOND)/	INDIP050
	. 1X,14HNUMBER I (KM),5X,4H(KM),4X,25H(M) (M) I(DEG K) (M/S),	INDIP051
	. 3X,10H(M) IFLAGI,4X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/	INDIP052
	. 1X,119(1H-))	INDIP053
	101 FORMAT(I6,1X,2F9.2,2F7.1,F7.0,2F7.1,F4.0,5(1PE11.3))	INDIP054
	110 FORMAT(1H0,A8,13H ARFA SOURCES/1X,94(1H-)/	INDIP055
	. 8X,1HI,11X,8HGEOMETRY,11X,1HI/	INDIP056
	. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,6H SIDE I,	INDIP057
	. 14X,28HEMISSIONS(MICROGRAMS/SECOND)/	INDIP058
	. 1X,14HNUMBER I (KM),5X,4H(KM),4X,3H(M),4X,5H(M) I,	INDIP059
	. 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/1X,94(1H-))	INDIP060
	111 FORMAT(I6,1X,2F9.2,2F7.1,5(1PE11.3))	INDIP061



120	FORMAT(1H0,A8,13H LINE SOURCES/1X,123(1H-)/	INDIP062
	. 8X,1HI,11X,8HGEOMETRY,11X,1HI,10X,1HI,54X,15HI AIRCRAFT ONLY/	INDIP063
	. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,3X,18HWIDTH I VELOCITY I,	INDIP064
	. 13X,28HEMISSIONS(MICROGRAMS/SECOND),13X,15HI LENGTH TIME/	INDIP065
	. 1X,14HNUMBER I (KM),5X,4H(KM),4X,23H(M) (M) I (KM/HR) I,	INDIP066
	. 5X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2,3X,15HI (KM) (HR)/	INDIP067
	. 1X,123(1H-))	INDIP068
121	FORMAT(I6,1X,2F9.2,2F7.1,6(1PE11.3),0PF7.2,1PE11.3)	INDIP069
122	FORMAT(7X,2F9.2,F7.1,7X,1PE11.3)	INDIP070
1200	FORMAT(1H0,A8,13H LINE SOURCES/1X,96(1H-)/	INDIP071
	. 8X,1HI,11X,8HGEOMETRY,12X,1HI/	INDIP072
	. 1X,8HSOURCE I,3X,1HX,8X,1HY,7X,1HZ,4X,7HWIDTH I,	INDIP073
	. 23X,28HEMISSIONS(MICROGRAMS/SECOND)/	INDIP074
	. 1X,14HNUMBER I (KM),5X,4H(KM),4X,3H(M),4X,6H(M) I,	INDIP075
	. 6X,2HCO,9X,2HHC,8X,3HNOX,9X,2HPT,8X,3HSO2/	INDIP076
	. 1X,96(1H-))	INDIP077
1211	FORMAT(I6,1X,2F9.2,2F7.1,2X,5(1PE11.3))	INDIP078
1222	FORMAT(7X,2F9.2,F7.1)	INDIP079
200	FORMAT(25HOWIND INDEPENDENT SOURCES/1H0)	INDIP080
	RETURN	INDIP081
C	ENTRY DEPINP	INDIP082
C		INDIP083
C	AT THIS ENTRY ALL WIND DEPENDENT SOURCES ARE PRINTED	INDIP084
C		INDIP085
	WRITE(6,300)WS,WD	INDIP086
300	FORMAT(1H1,'WIND DEPENDENT SOURCES FOR',F8.4,' MPS WIND SPEED AND'	INDIP087
	. ,F8.4,' RADIANS WIND DIRECTION')	INDIP088
	IF (IFLAG.EQ.0) GO TO 18	INDIP089
	IF (NACPT.EQ.0) GO TO 17	INDIP090
	WRITE(6,100) SORNAM(3)	INDIP091
	DC 7 I=1,NACPT	INDIP092
	7 WRITE(6,101) I, (ACPT(J,I),J=1,4), (ACPT(J,I),J=6,8), (ACPT(J,I),	INDIP093
	. J=11,15)	INDIP094
17	IF(NACAR.EQ.0) GO TO 18	INDIP095
	WRITE(6,110) SORNAM(3)	INDIP096
	DC 8 I=1,NACAR	INDIP097
	8 WRITE(6,111) I, (ACAR(J,I),J=1,4), (ACAR(J,I),J=6,10)	INDIP098
18	IF(NACLN.EQ.0) GO TO 19	INDIP099
	WRITE(6,120) SORNAM(3)	INDIP100
	DC 9 I=1,NACLN	INDIP101
	IF (ACLN(9,I).NE.1.0) GO TO 1987	INDIP102
	WRITE(6,1219) I, (ACLN(J,I),J=1,4), (ACLN(J,I),J=13,17), ACLN(11,I)	INDIP103
	WRITE(6,1229) (ACLN(J,I),J=6,8)	INDIP104
1219	FORMAT(1X,1X,2F9.2,2F7.1,4X,3HN/A,4X,5(1PE11.3),0PF7.1,4X,3HN/A)	INDIP105
1229	FORMAT(7X,2F9.2,F7.1,11X,3HN/A)	INDIP106
	GC TC 9	INDIP107
1987	CONTINUE	INDIP108
	WRITE(6,121) I, (ACLN(J,I),J=1,4), ACLN(9,I), (ACLN(J,I),J=13,17),	INDIP109
	. ACLN(11,I),ACLN(12,I)	INDIP110
	WRITE(6,122) (ACLN(J,I),J=6,8),ACLN(10,I)	INDIP111
9	CONTINUE	INDIP112
19	CONTINUE	INDIP113
	RETURN	INDIP114
	END	INDIP115
		INDIP116

## PROGRAM MAIN

### Purpose:

To read the general problem input, set up the receptor grid, call a routine to read the master emission file and then call the short-term model.

### Input:

1. Problem title
2. Definition of pollutants to be output
3. Description of special cases
4. Description of receptor grid
5. Description of statistical receptors

### Output:

All input is printed

### Procedure:

1. Read card input
2. Calculate receptor locations
3. Check statistical receptors against the receptor locations
4. Call routine to read master emission file
5. Call the short-term model

### Subroutines Called:

READ, MAINS

AD-A046 348

ARGONNE NATIONAL LAB ILL  
AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)  
APR 77 D J BINGAMAN

F/G 13/2

UNCLASSIFIED

CEEDO-TR-76-34

NL

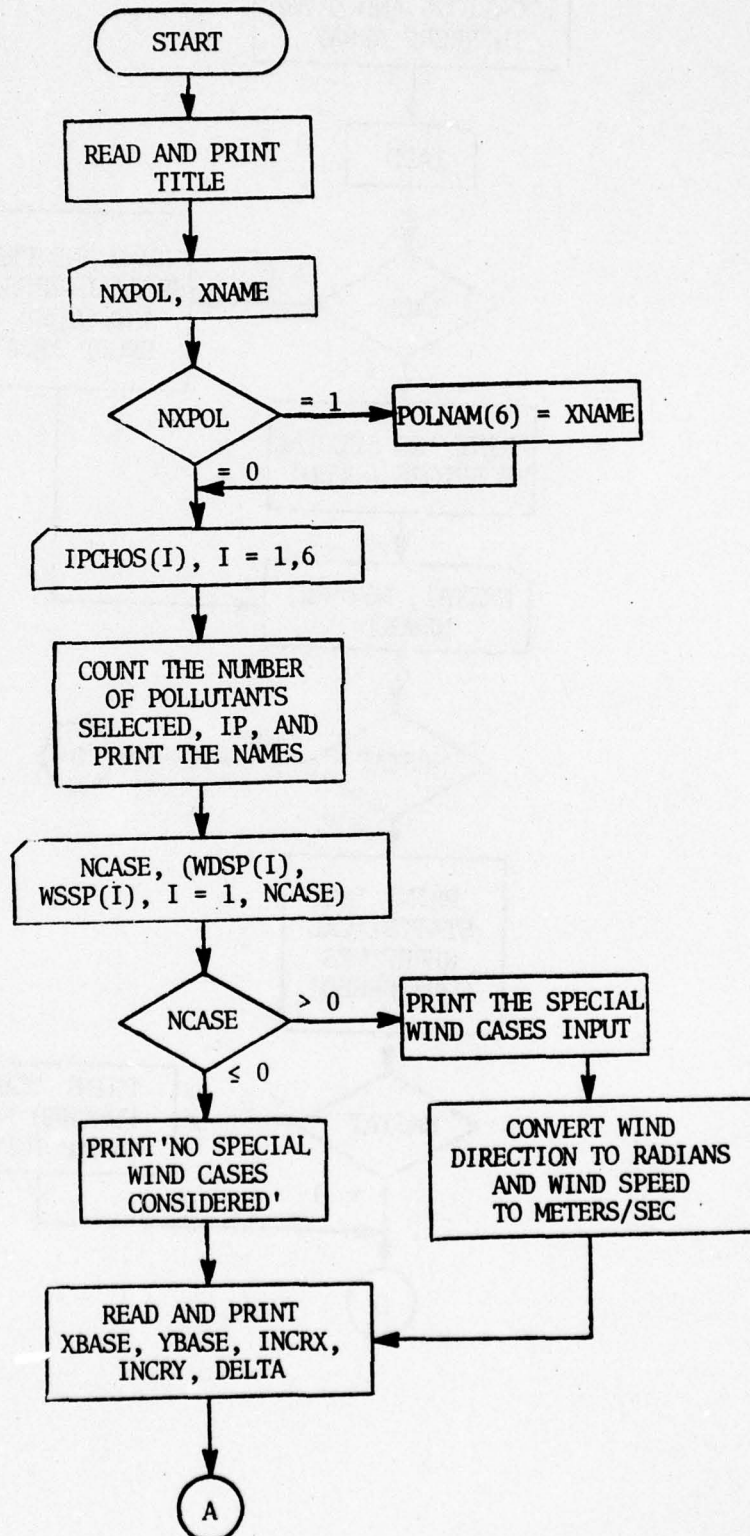
2 of 3

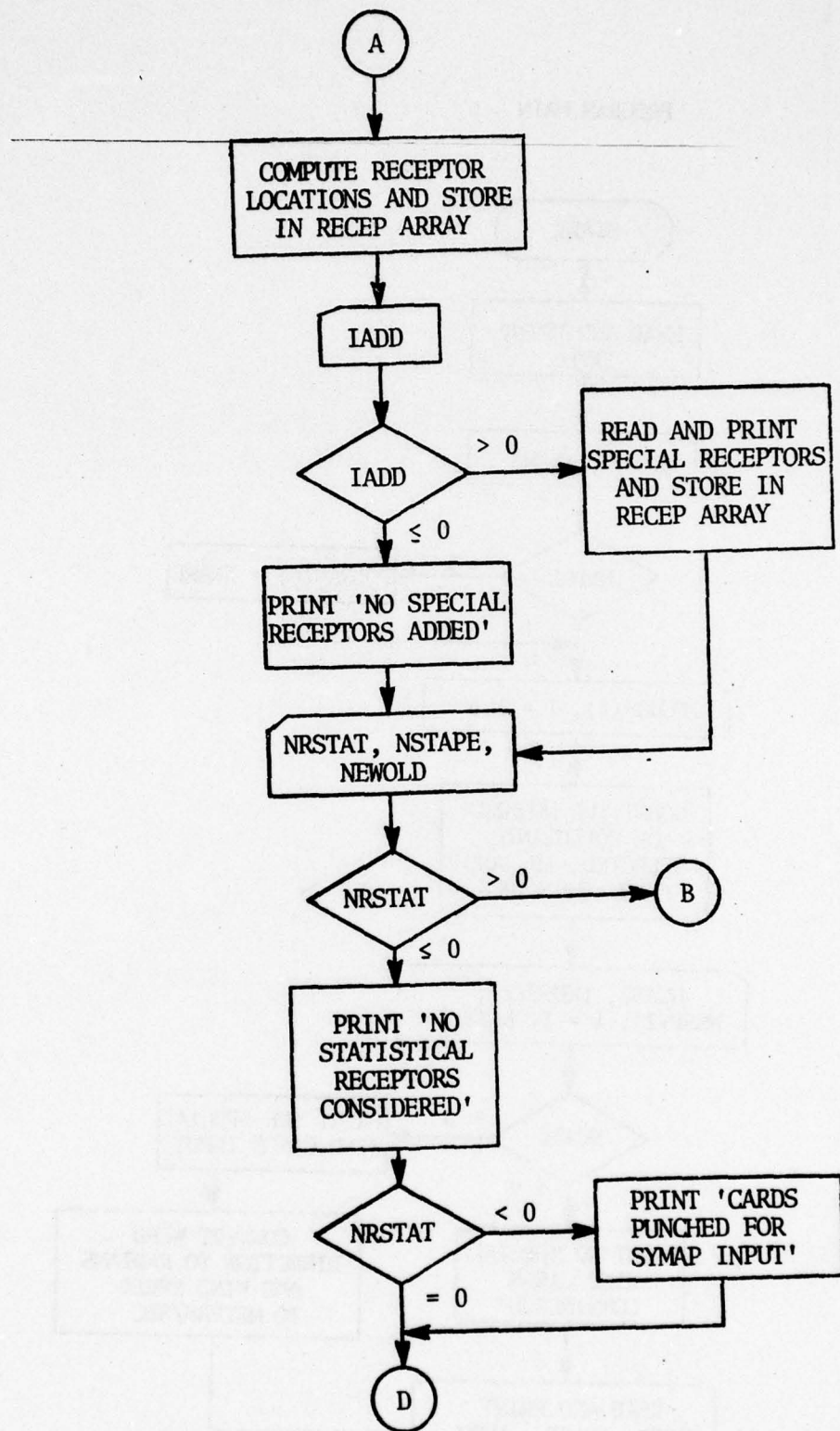
AD  
A046348

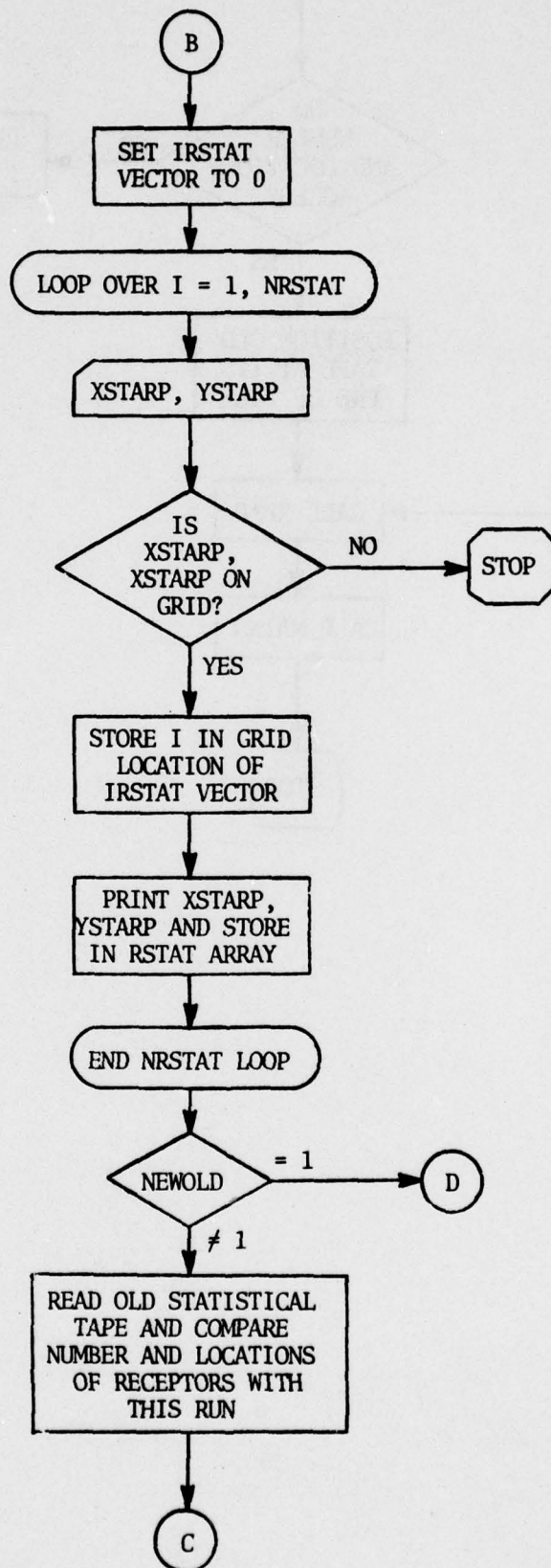




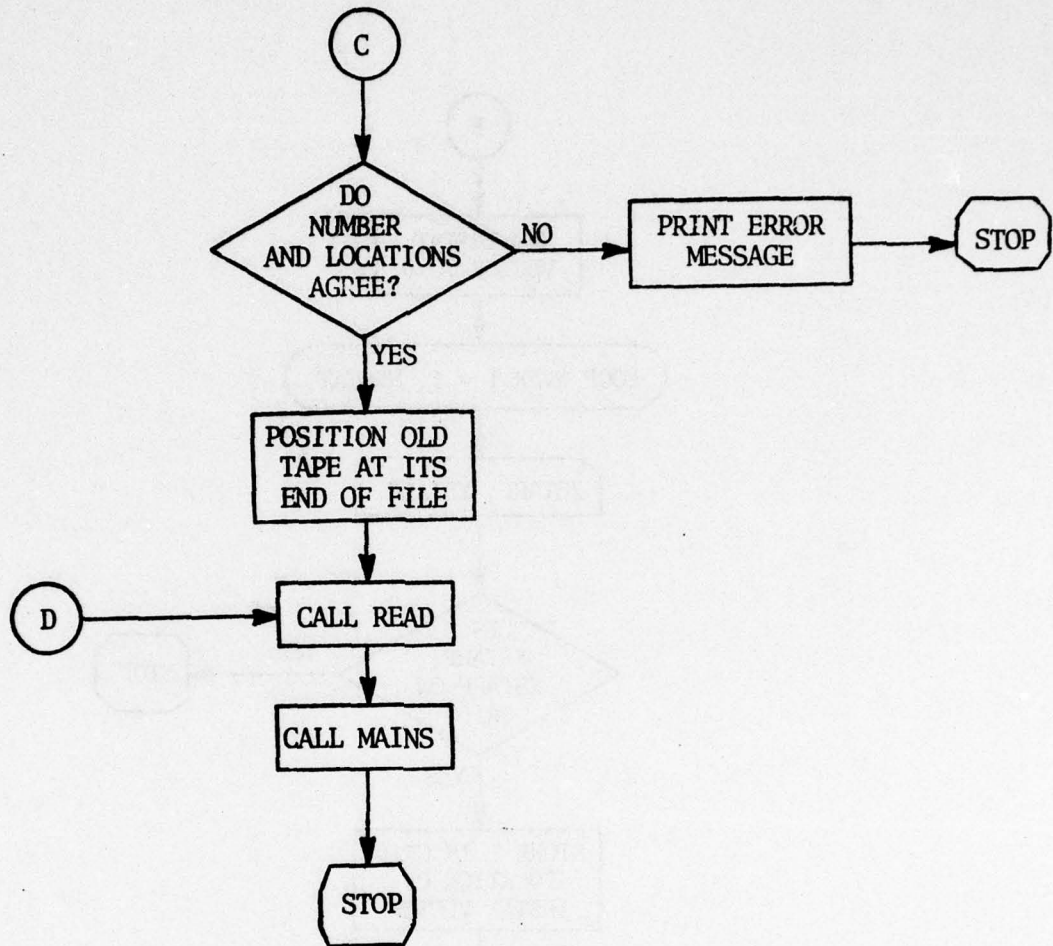
# PROGRAM MAIN











C		MAIN0000
C	THIS PROGRAM IS THE MAIN DRIVER ROUTINE WHICH READS IN RECEPTOR AND	MAIN0001
C	OTHER GENERAL DATA, CALLS SUBROUTINE READ TO READ THE MASTER	MAIN0002
C	SOURCE EMISSION TAPE, AND THE DIRECTS CONTROL TO MAINS FOR THE	MAIN0003
C	SHORT TERM MODEL	MAIN0004
C		MAIN0005
	REAL*8 POLNAM,XNAME	MAIN0006
	COMMON /AIRQAL/ RECDAT(3, 6,312)	MAIN0007
	COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR	MAIN0008
	COMMON /INPC/ IRECEP,IWDIR,ITYPE,HTAERO,SORC(18),IPOL	MAIN0009
	COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP	MAIN0010
	,TEMK,UA	MAIN0011
	COMMON /MCNMET/ TMPAR,WSHMBAR,AMDMBR,DTMBAR	MAIN0012
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG,IONCE	MAIN0013
	COMMON /RCPT/ NRECEP,RECEP(2,312)	MAIN0014
	COMMON /SPEC/ NCASE,WSSP(3),WDSP(3)	MAIN0015
	COMMON /SRCE/ NPOL,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,	MAIN0016
	. NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),	MAIN0017
	. ABAR(11,100),ASLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)	MAIN0018
	COMMON /TITL/ POLNAM(6),TITLE1(20),IPCHOS(6),NXPOL,IP	MAIN0019
	COMMON /STAT/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)	MAIN0020
	DIMENSION AA (20),OSTATR(2,20)	MAIN0021
C		MAIN0022
C	READ AND PRINT RECEPTOR AND OTHER GENERAL INPUT	MAIN0023
C		MAIN0024
	1 READ(5,100) TITLE1	MAIN0025
	100 FCRMAT(20A4)	MAIN0026
	PRINT 200, TITLE1	MAIN0027
	200 FORMAT(1H1,20A4)	MAIN0028
	REAL(5,110) NXPOL, XNAME	MAIN0029
	110 FCRMAT(16,5A8)	MAIN0030
	IF(NXPOL.EQ.0) GO TO 31	MAIN0031
	ECLNAM(6)=XNAME	MAIN0032
	31 CONTINUE	MAIN0033
	READ(5,130) (IPCHOS(I),I=1,6)	MAIN0034
	130 FCRMAT(10I6)	MAIN0035
	DO 40 I=1,6	MAIN0036
	IF(IPCHOS(I).IF.0) GO TO 41	MAIN0037
	40 CONTINUE	MAIN0038
	41 IF=I-1	MAIN0039
	PRINT 203, (POLNAM(IPCHOS(I)),I=1,IP)	MAIN0040
	203 FORMAT(21H0POLLUTANTS SELECTED /6A8)	MAIN0041
	READ(5,140) NCASE,(WDSP(I),WSSP(I),I=1,NCASE)	MAIN0042
	140 FORMAT(16,6F6.0)	MAIN0043
	IF (NCASE) 48,48,49	MAIN0044
	48 PRINT 201	MAIN0045
	201 FORMAT(33H00 SPECIAL WIND CASES CONSIDERED)	MAIN0046
	GC TC 51	MAIN0047
	49 PRINT 202, (I,WDSP(I),WSSP(I),I=1,NCASE)	MAIN0048
	202 FCRMAT(20H0SPECIAL WIND CASES /53H CASE WIND DIRECTION(DEGREES	MAIN0049
	.) WIND SPEED(KNOTS)/(16,F18.2,F23.2))	MAIN0050
	DC 50 I=1,NCASE	MAIN0051
	WDSP(I)=WDSP(I)*0.0174533	MAIN0052
	50 WSSP(I)=WSSP(I)*0.5148	MAIN0053
	51 CCNTINUE	MAIN0054
	READ(5,120) XBASE,YBASE,INCRX,INCRY,DELTA	MAIN0055
	120 FORMAT(2F8.0,2I8,F8.0)	MAIN0056
	PRINT 204, XBASE,YBASE,INCRX,INCRY,DELTA	MAIN0057
	204 FORMAT(43H0LOWER LEFT CORNER OF RECEPTOR GRID IS AT (,F8.3,1H,,	MAIN0058
	.F8.3,1H)/12H THERE ARE,14,12H COLUMNS AND,14,23H ROWS WITH A SPAN	MAIN0059
	.CING OF,F6.2,11H KILOMETERS)	MAIN0060
	NRECEP=0	MAIN0061

DO 10 I=1,INCPY	MAIN0062
DC 10 J=1,INCRY	MAIN0063
NRECEP=NRECEP+1	MAIN0064
RECEP(1,NRECEP)=XBASE+(I-1)*DELTA	MAIN0065
10 RECEP(2,NRECEP)=YBASE+(J-1)*DELTA	MAIN0066
READ(5,110)IADD	MAIN0067
IF (IADD) 14,14,15	MAIN0068
14 PRINT 205	MAIN0069
205 FORMAT(27H0NO SPECIAL RECEPTORS ADDED)	MAIN0070
GC TC 21	MAIN0071
15 PRINT 206	MAIN0072
206 FORMAT(25H0SPECIAL RECEPTORS ADDED /36H NO. X-COORDINATE Y-C	MAIN0073
.OCDINATE)	MAIN0074
DO 20 I=1,IADD	MAIN0075
READ(5,120)XRECEP,YRECEP	MAIN0076
NRECEP=NRECEP+1	MAIN0077
PRINT 207, NRECEP,XRECEP,YRECEP	MAIN0078
207 FORMAT(I5,F14.3,F15.3)	MAIN0079
RECEP(1,NRECEP)=XRECEP	MAIN0080
RECEP(2,NRECEP)=YRECEP	MAIN0081
20 CCNTINUE	MAIN0082
21 CONTINUE	MAIN0083
C	MAIN0084
C READ AND PRINT STATISTICAL RECEPTOR INPUT	MAIN0085
C	MAIN0086
READ 130,NRSTAT,NSTAPE,NEWOLD	MAIN0087
IF (NRSTAT.GT.0) GO TC 305	MAIN0088
PRINT 302	MAIN0089
302 FORMAT(36H0NO STATISTICAL RECEPTORS CONSIDERED)	MAIN0090
IF (NRSTAT.LT.0) PRINT 303	MAIN0091
303 FCRMAI(30H0CAPDS PUNCHED FOR SYMAP INPUT)	MAIN0092
GO TC 400	MAIN0093
305 PRINT 301,NRSTAT	MAIN0094
301 FORMAT(1H0,I8,22H STATISTICAL RECEPTORS)	MAIN0095
DO 310 I=1,NRECEP	MAIN0096
IRSTAT(I)=0	MAIN0097
310 CONTINUE	MAIN0098
DC 340 I=1,NRSTAT	MAIN0099
READ 120,XSTARP,YSTARP	MAIN0100
DO 320 IC=1,NRECEP	MAIN0101
IF (XSTARP.EQ.RECEP(1,IC).AND.YSTARP.EQ.RECEP(2,IC)) GO TO 330	MAIN0102
320 CCNTINUE	MAIN0103
PRINT 321,XSTARP,YSTARP	MAIN0104
321 FORMAT(25H0STATISTICAL RECEPTOR X =,F7.3,5H, Y =,F7.3,	MAIN0105
. 12H NOT ON GRID)	MAIN0106
STOP	MAIN0107
330 ISTAT(IC)=I	MAIN0108
PRINT 322,I,XSTARP,YSTARP	MAIN0109
322 FORMAT(I12,7H AT X =,F10.3,4H Y =,F10.3)	MAIN0110
RSTAT(1,I)=XSTARP	MAIN0111
RSTAT(2,I)=YSTARP	MAIN0112
340 CCNTINUE	MAIN0113
IF (NEWOLD.EQ.1) GO TO 400	MAIN0114
READ (NSTAPE) AA,IH,JRSTAT,((OSTATR(I,J),I=1,2),J=1,JRSTAT),	MAIN0115
. JWE,WDJ,JWS,WSJ,JJJ,HLIDJ	MAIN0116
IF (JRSTAT.EQ.NRSTAT) GO TO 350	MAIN0117
PRINT 341	MAIN0118
341 FORMAT(46H0NUMBER OF STATISTICAL RECEPTORS ON OLD TAPE (,I2,	MAIN0119
. 42H), DO NOT AGREE WITH NUMBER FOR THIS RUN (,I2,1H))	MAIN0120
STOP	MAIN0121
350 DC 360 J=1,NRSTAT	MAIN0122
DC 360 I=1,2	MAIN0123



IF (CSTAIR(I,J).EQ.RSTAT(I,J)) GO TO 360	MAIN0124
PRINT 351,((OSTATR(I,J),I=1,2),J=1,NRSTAT)	MAIN0125
351 FCRMAT(64H0STATISTICAL RECEPTOR LOCATIONS DO NOT MATCH,THOSE ON TA	MAIN0126
.FE ARE ,20(/2F15.3))	MAIN0127
STOP	MAIN0128
360 CONTINUE	MAIN0129
C	MAIN0130
C POSITION OLD TAPE AT END OF LAST RECORD	MAIN0131
C	MAIN0132
370 READ (NSTAPE,END=380)	MAIN0133
GO TO 370	MAIN0134
380 BACKSPACE NSTAPE	MAIN0135
BACKSPACE NSTAPE	MAIN0136
400 CCNTINUE	MAIN0137
CALL READ	MAIN0138
CALL MAINS	MAIN0139
STOP	MAIN0140
RETURN	MAIN0141
END	MAIN0142

## SUBROUTINE MAINS

### Purpose:

To direct the short term calculation by reading the data, checking for special cases and calling the source and diffusion calculation routines.

### Input:

1. Card input to describe the time periods to be calculated
2. Card input to describe the meteorological conditions
3. Special case data from the MAIN routine

### Output:

1. Common block data to be used by the calculation and output subroutines
2. Statistical receptor data on tape and/or cards for SYMAP

### Procedure:

1. Set constants
2. Read general data
3. For each period:
  - a. Read time and meteorological data
  - b. Check for near zero wind speed
  - c. Check limits of mixing depth
  - d. For each hour:
    1. Find the critical distance
    2. Set wind direction and speed classes
    3. Call the non-aircraft source routines
    4. Check for special cases of wind speed and direction
    5. Call the aircraft source routines
    6. Call the diffusion model and output routines
    7. Check for statistical output, including cards for SYMAP.

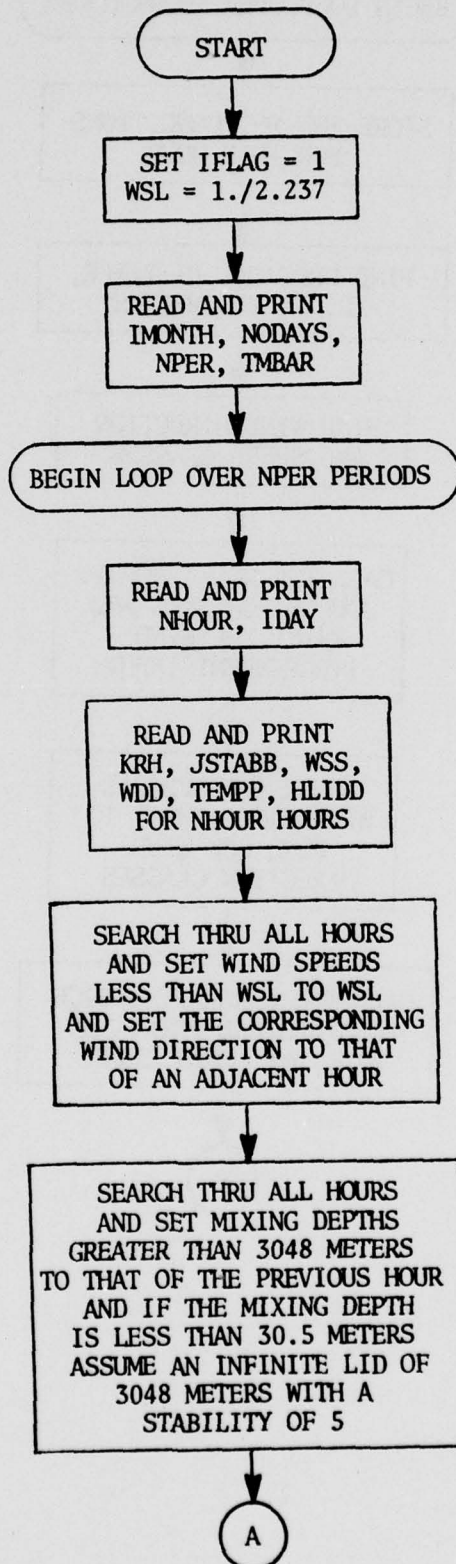
### Subroutines Called:

SOURCE, INDINP, ACSRCE, DEPINP, POLSOR, OUTPUT

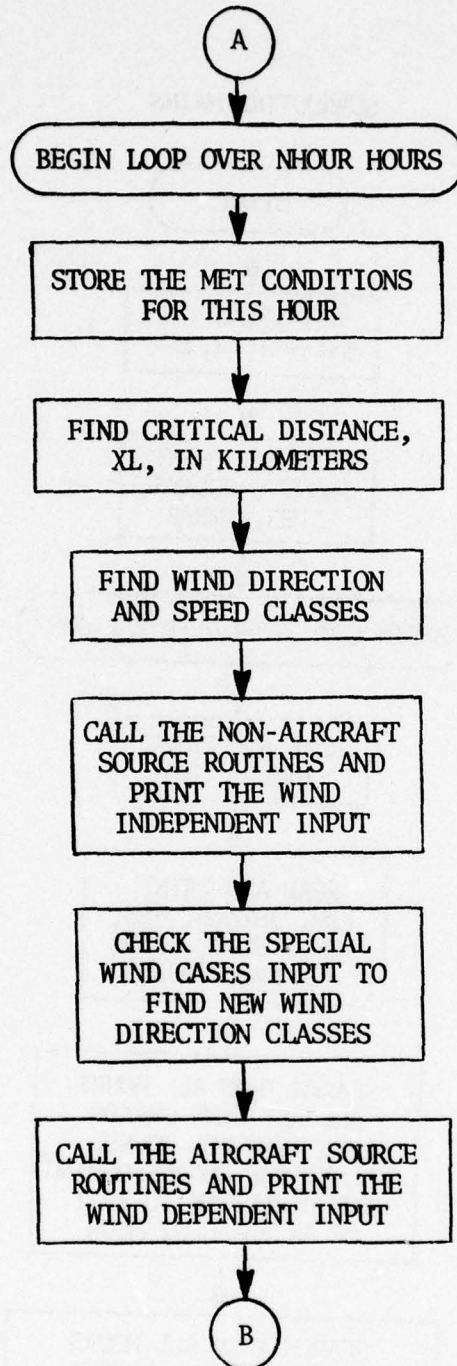
### Function Called:

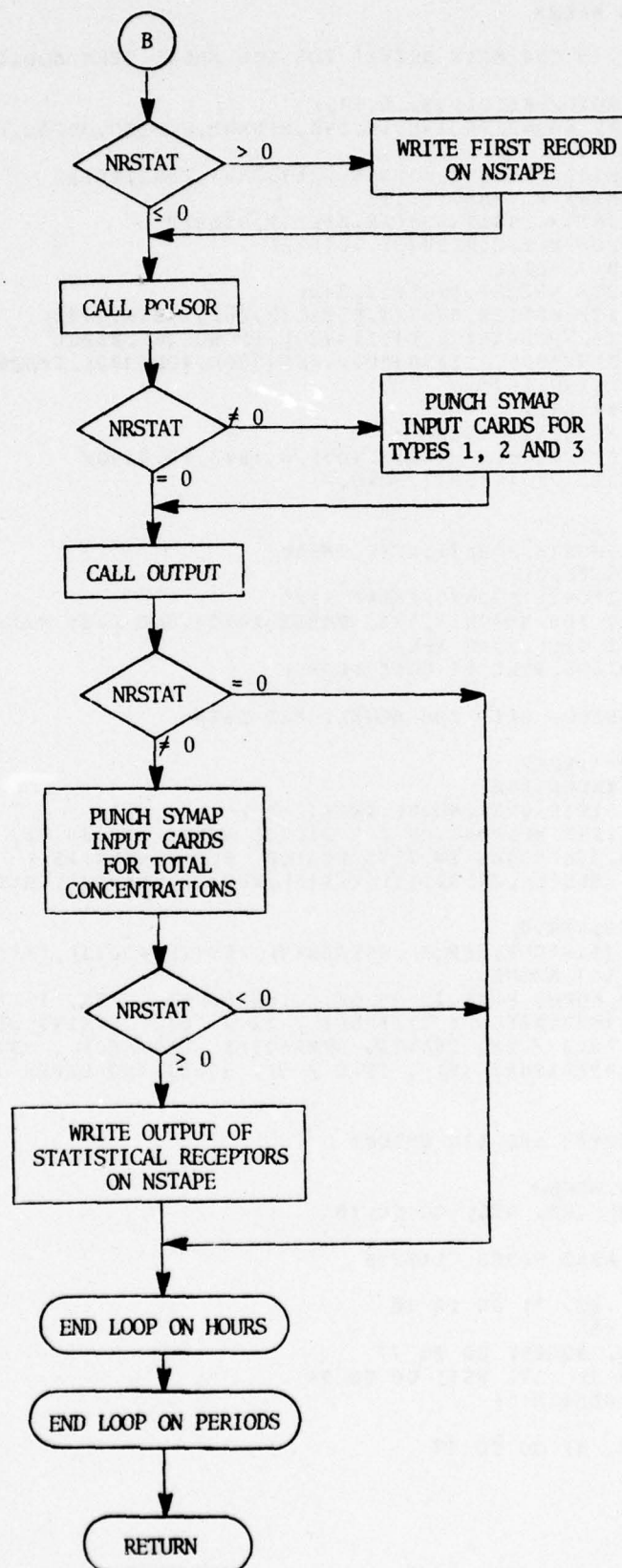
SIGCZ

SUBROUTINE MAINS









	SUBROUTINE MAINS	MAINS000
C		MAINS001
C	THIS ROUTINE IS THE MAIN DRIVER FOR THE SHORT TERM MODEL	MAINS002
C		MAINS003
	COMMON /AIRQAL/ RECDAT(3, 6,312)	MAINS004
	COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,	MAINS005
	. IEMK	MAINS006
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG	MAINS007
	CCMMCN /XTRAN/ XL,WSMD,TY,TZ	MAINS008
	COMMON /MONMET/ TMBAR,WSMBAR,AMDMBR,DIMBAR	MAINS009
	CCMMCN /SPEC/ NCASE,WSSP(3),WDSP(3)	MAINS010
	COMMON /WDUN/ WSAVE	MAINS011
	COMMON /RCPT/ NRECEP,RECEP(2,312)	MAINS012
	COMMON /STAI/ NSTAPE,NRSTAT,RSTAT(2,20),IRSTAT(312)	MAINS013
	COMMON /TITL/ PCLNAM(6),TITLE1(20),IPCHOS(6),NXPOL	MAINS014
	DIMENSION KRH(100),JSTABB(100),WSS(100),WDD(100),TEMPP(100),	MAINS015
	. HLIDD(100),WSCLAS(5)	MAINS016
	DIMENSION WEEK(2)	MAINS017
	DATA WEEK / 3HDAY, 3HEND /	MAINS018
	DATA WSCLAS/1.8018,3.3462,5.4055,8.4943,10.8110/	MAINS019
	DATA PI,SHLID /3.1415927,3048./	MAINS020
	IFLAG = 1	MAINS021
	WSL = 1./2.237	MAINS022
	READ 100, IMONTH,NODAYS,NPER,TMBAR	MAINS023
100	FCRMAI(316,F6.0)	MAINS024
	PRINT 203,IMONTH,NODAYS,TMBAR,NPER	MAINS025
203	FCRMAI(11H0 FOR MONTHI3,11H, THERE AREI4,36H DAYS WITH AN AVERAGE	MAINS026
	.TEMPERATURE OFF6.2,4H (F)/	MAINS027
	.I5,27H PERIODS WILL BE CONSIDERED)	MAINS028
C		MAINS029
C	FOR EVERY PERIOD, READ THE HOURLY MET DATA	MAINS030
C		MAINS031
	DO 50 IPER=1,NPER	MAINS032
	READ 100, NHOURL,IDAY	MAINS033
	PRINT 201, IPER,NPER,NHOURL,WEEK(IDAY)	MAINS034
201	FCRMAI(1H1,29HINFORMATION FOR PERIOD NUMBER,I3,3H OF, I3,8H PERIOD	MAINS035
	.S / 5X,I3,33H HOURS IN THIS PERIOD, FOR A WEEK,A3 )	MAINS036
	READ 101, (KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),HLIDD(I),	MAINS037
	-I=1,NHOURL)	MAINS038
101	FORMAT (2I6,4F6.0)	MAINS039
	PRINT 202, (I,NHOURL,KRH(I),JSTABB(I),WSS(I),WDD(I),TEMPP(I),	MAINS040
	. HLIDD(I),I=1,NHOURL)	MAINS041
202	FORMAI(1H0,8HFOR HOUR,I5,3H OF,I5,6H HOURS /5X, 10H HOUR INDEX ,	MAINS042
	. I3 / 5X, 18HSTABILITY CATEGORY , I2 / 5X, 26HWIND SPEED (METERS/	MAINS043
	.SECONDE) , F8.2 / 5X, 24HWIND DIRECTION (DEGREES) , F8.2 /	MAINS044
	. 5X, 15HTEMPERATURE (F) , F8.2 / 5X, 21HMIXING DEPTH (METERS) ,	MAINS045
	. F 8.2 )	MAINS046
C		MAINS047
C	CHECK WIND SPEED AND LID HEIGHT	MAINS048
C		MAINS049
	DO 20 IH=1,NHOURL	MAINS050
	IF (WSS(IH) .GE. WSL) GO TO 18	MAINS051
C		MAINS052
C	REMOVE ZERO WIND SPEED CLASSES	MAINS053
C		MAINS054
	IF (NHOURL .EQ. 1) GO TO 16	MAINS055
	WSS(IH) = WSL	MAINS056
	IF (IH .EQ. NHOURL) GO TO 17	MAINS057
	IF (WSS(IH+1) .LT. WSL) GO TO 14	MAINS058
	WDD(IH) = WDD(IH+1)	MAINS059
	GO TO 18	MAINS060
14	IF (IH .NE. 1) GO TO 17	MAINS061



DC 15 JH=2,NHOUR	MAINS062
IF (WSS(JH) .LT. WSL) GO TO 15	MAINS063
WDD(IH)= WDD(JH)	MAINS064
GO TO 18	MAINS065
15 CCNTINUE	MAINS066
16 PRINT 200	MAINS067
STOF	MAINS068
200 FCRMAT(1H1,10X,'WINDSPEED IS TOO SMALL')	MAINS069
17 WDD(IH) = WDD(IH-1)	MAINS070
C	MAINS071
C SET INFINITE LID HEIGHT AT 3048 METERS	MAINS072
C	MAINS073
18 IF (HLIDD(IH).GT. 3048.) HLIDD(IH) = SHLID	MAINS074
IF (HLIDD(IH) .GE. 30.5) GO TO 19	MAINS075
C	MAINS076
C GROUND LEVEL INVERSION, ASSUME INFINITE LID WITH A STABILTY OF 5	MAINS077
C	MAINS078
HLIDD(IH) = 3048.	MAINS079
JSTABB(IH) = 5	MAINS080
19 SHLID = HLIDD(IH)	MAINS081
20 CCNTINUE	MAINS082
C	MAINS083
C BEGIN HCURLY CALCULATIONS	MAINS084
C	MAINS085
DO 30 IHOURL = 1,NHOUR	MAINS086
IHR1 = KRH(IHCUR)	MAINS087
WS = WSS(IHOURL)	MAINS088
IHP2 = IHR1	MAINS089
WD = WDD(IHOURL)*0.0174533	MAINS090
TEMP = TEMPP(IHOURL)	MAINS091
TEMK = (TEMP-32.)/1.8 + 273.	MAINS092
JSTAB = JSTABB(IHOURL)	MAINS093
HLID = HLIDD(IHOURL)	MAINS094
WSMPH = WS * 2.237	MAINS095
SINEWD = SIN(WD)	MAINS096
COSEWD = COS(WD)	MAINS097
HL = 0.47*HLID	MAINS098
WSAVE=WS	MAINS099
IF (HL.LT.1.0) HL=1.0	MAINS100
C	MAINS101
C FIND CRITICAL DISTANCE, XL, IN KILOMETERS	MAINS102
C	MAINS103
XL=SIGCZ(JSTAB,HL)/1000.	MAINS104
C	MAINS105
C FIND WIND DIRECTION AND SPEED CLASSES	MAINS106
C	MAINS107
DC 21 K=1,16	MAINS108
IF (WD .GT. (22.5*K-11.25)*PI/180.) GO TO 21	MAINS109
IWD = K	MAINS110
GC TC 22	MAINS111
21 CCNTINUE	MAINS112
IWD=1	MAINS113
22 DC 23 K=1,5	MAINS114
IF (WS.GT.WSCLAS(K)) GO TO 23	MAINS115
IWS = K	MAINS116
GO TO 25	MAINS117
23 CCNTINUE	MAINS118
IWS = 6	MAINS119
25 CCNTINUE	MAINS120
PRINT 925,IHOURL	MAINS121
925 FORMAT(9H1FOR HOUR13)	MAINS122
C	MAINS123

C	CALL THE MCN-AIRCRAFT SOURCE ROUTINES AND PRINT THE	MAINS124
C	WIND INDEPENDENT INPUT	MAINS125
C	CALL SCURCE	MAINS126
	CALL INDINP	MAINS127
	IWD = IWD	MAINS128
	IF (NCASE .EQ. 0) GO TO 28	MAINS129
	DC 26 I=1, NCASE	MAINS130
	COMP = WS * COS(WD - WDSP(I))	MAINS131
	IF (COMP .LT. WSSP(I)) GO TO 26	MAINS132
	IWD = 17 + I	MAINS133
	GO TO 28	MAINS134
26	CCONTINUE	MAINS135
28	CONTINUE	MAINS136
C		MAINS137
C	CALL THE AIRCRAFT SOURCE ROUTINES AND PRINT THE	MAINS138
C	WIND DEPENDENT INPUT	MAINS139
C		MAINS140
	CALL ACSRCE	MAINS141
	CALL LEPINP	MAINS142
	IWD = IWD	MAINS143
C		MAINS144
C	IF STATISICAL OPTION IS CHOSEN, WRITE FIRST RECORD ON NSTAPE	MAINS145
C		MAINS146
	IF (NRSTAT.LE.0) GO TO 300	MAINS147
	NHR=1	MAINS148
	WRITE (NSTAPE) TITLE1,IHR1,NRSTAT,((RSTAT(I,J),I=1,2),J=1,NRSTAT),	MAINS149
	. IWD,WD,IWS,WS,JSTAB,HLID,NHR	MAINS150
300	CONTINUE	MAINS151
C		MAINS152
C	CALL THE DIFFUSION MODEL	MAINS153
C		MAINS154
	CALL FCISOR	MAINS155
	IF (NRSTAT.EQ.0) GO TO 320	MAINS156
	DC 310 K=1,3	MAINS157
	PUNCH 301,NRECEP,K	MAINS158
301	FORMAT(2I6)	MAINS159
	DC 310 N=1,NRECEP	MAINS160
	PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(K,J,N),J=1,6)	MAINS161
302	FORMAT(1P8E10.3)	MAINS162
310	CCONTINUE	MAINS163
320	CONTINUE	MAINS164
C		MAINS165
C	PRINT RESULTS	MAINS166
C		MAINS167
	CALL CUPUT	MAINS168
C		MAINS169
	IF THE STATISTICAL OPTION IS CHOSEN, RECORD THE OUTPUT	MAINS170
		MAINS171
	IF (NRSTAT.EQ.0) GO TO 360	MAINS172
	K=4	MAINS173
	PUNCH 301,NRECEP,K	MAINS174
	DO 330 N=1,NRECEP	MAINS175
	PUNCH 302,(RECEP(I,N),I=1,2),(RECDAT(1,J,N),J=1,6)	MAINS176
330	CCONTINUE	MAINS177
	IF (NRSTAT.LT.0) GO TO 360	MAINS178
	K=0	MAINS179
	DC 350 N=1,NRECEP	MAINS180
	IF (IRSTAT(N).LE.0) GO TO 350	MAINS181
	K=K+1	MAINS182
	DO 340 J=1,6	MAINS183
	RECDAT(1,J,K)=RECDAT(1,J,N)	MAINS184
		MAINS185

340 CCNTINUE  
350 CCNTINUE  
WRITE (NSTAPE) ((RECDAT(1,J,N),J=1,6),N=1,NRSTAT)  
360 CCNTINUE  
30 CONTINUE  
50 CCNTINUE  
RETURN  
END

MAINS186  
MAINS187  
MAINS188  
MAINS189  
MAINS190  
MAINS191  
MAINS192  
MAINS193



## SUBROUTINE METHA

### Purpose:

To calculate diurnal emissions allowing each source in a class to have a unique or default distribution pattern.

### Input:

1. The ICLASS number of the sources and NPTC, the number of sources not using the default of a uniform distribution.
2. For each of the NPTC sources, the source ID number and fractions of the hour, day and month, FH, FD and FM, which that source is on. If one or two of the fractions are left blank, the default is used. If all are blank, the source is assumed to be off.

### Output:

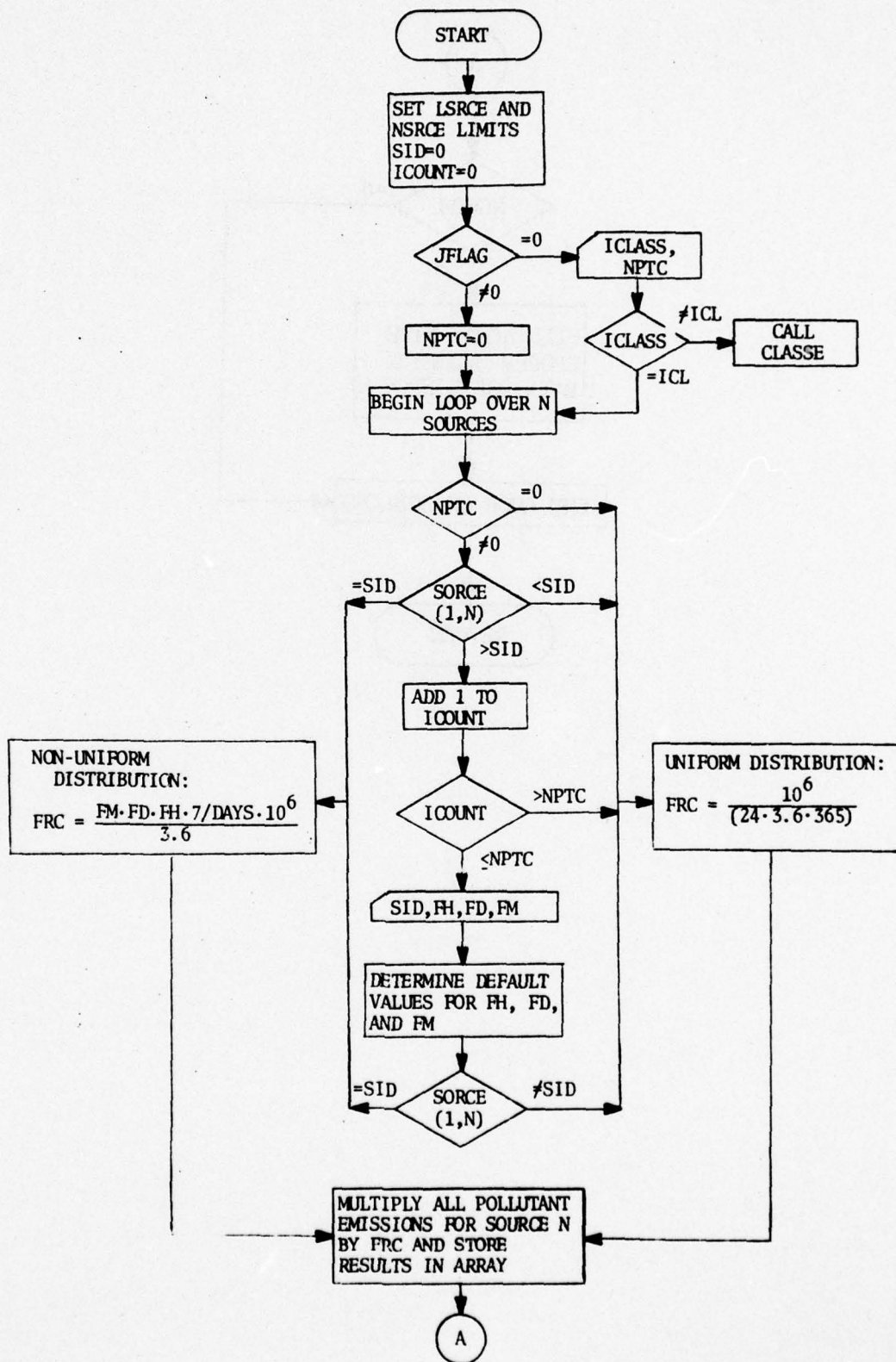
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines

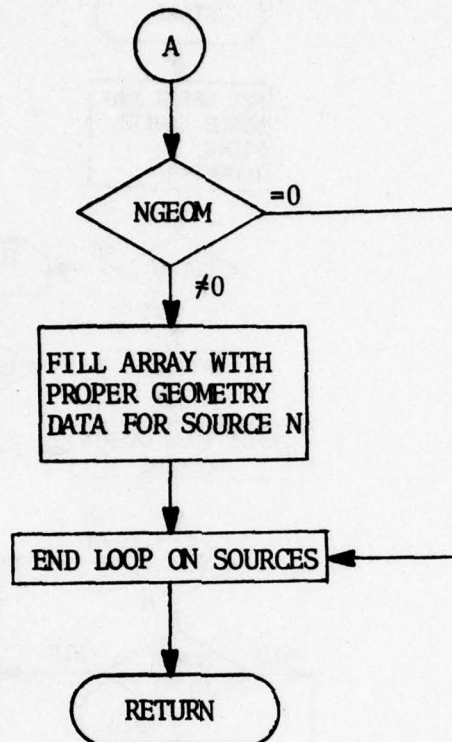
#### Called:

CLASSE

# SUBROUTINE METHA



SUBROUTINE METHA (Contd.)





	SUBROUTINE METHA(MAXN,ARRAY,I1,I2,ICL)	METHA000
C		METHA001
C	THIS ROUTINE CALCULATES DIURNAL EMISSIONS ALLOWING EACH	METHA002
C	SOURCE IN A CLASS TO HAVE A DIFFERENT DISTRIBUTION PATTERN.	METHA003
C	DEFAULTS ARE: FH = 1/24	METHA004
C	FD = 1/7	METHA005
C	FM = 1/12 OR 1	METHA006
C		METHA007
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG	METHA008
	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,	METHA009
	. NACPT,NACAR,NENLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),	METHA010
	. ABET(16,150),ABAR(11,100),ABLN(14,100)	METHA011
	COMMON/JUNK/DA SRCE,NSRCE,SORCE(17,300),SORGM(10,200)	METHA012
	. ;LOC1,LOC2,NGEOM,NPT	METHA013
	DIMENSION ARRAY(I1,I2)	METHA014
	LSRCE=NSRCE+1	METHA015
	NSRCE=NSRCE+MAXN	METHA016
	SID=0.	METHA017
	ICOUNT=0	METHA018
	IF (JFLAG.EQ.0) GO TO 5	METHA019
	NPTC=0	METHA020
	GO TO 6	METHA021
5	READ 1,ICLASS,NPTC	METHA022
1	FORMAT(2I4)	METHA023
	IF (ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)	METHA024
6	DO 100 N=LSRCE,NSRCE	METHA025
	IF (NPTC.EQ.0) GO TO 30	METHA026
	IF (SID-SORCE(1,N)) 10,40,30	METHA027
10	ICOUNT=ICOUNT+1	METHA028
	IF (ICOUNT.GT.NPTC) GO TO 30	METHA029
	READ 2,SID,FH,FD,FM	METHA030
2	FORMAT(F4.0,4X,3F8.7)	METHA031
	IF (FH+FD+FM.EQ.0.0) GO TO 20	METHA032
C		METHA033
C	DETERMINE DEFAULT VALUES	METHA034
C		METHA035
	IF (FM.NE.0.0) GO TO 15	METHA036
	FM=1./12.	METHA037
	IF (DAYS.GE.365.) FM=1.	METHA038
15	IF (FD.EQ.0.0) FD=1./7.	METHA039
	IF (FH.EQ.0.0) FH=1./24.	METHA040
C		METHA041
20	CONTINUE	METHA042
	IF (SID-SORCE(1,N)) 30,40,30	METHA043
C		METHA044
C	UNIFORM DISTRIBUTION	METHA045
C		METHA046
30	FRC=1.CE+6/(24.*3.6*365.)	METHA047
	GO TO 50	METHA048
C		METHA049
C	NCN-UNIFORM DISTRIBUTION	METHA050
C		METHA051
40	FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)	METHA052
50	DO 60 I=1,NPLTS	METHA053
	ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC	METHA054
60	CONTINUE	METHA055
	IF (NGEOM.EQ.0) GO TO 100	METHA056
	DO 70 I=1,NGPOM	METHA057
	ARRAY(I,N)=SORCE(I+2,N)	METHA058
70	CONTINUE	METHA059
	IF (IFT.EQ.1) ARRAY(10,N)=SORCE(2,N)	METHA060
100	CONTINUE	METHA061

RETURN  
END

METHA062  
METHA063

## SUBROUTINE METHB

### Purpose:

To calculate diurnal emissions using a degree-hour method.

### Input:

The ICLASS number of the sources and UNIFRC, the fraction of emissions which are to be uniformly distributed.

### Output:

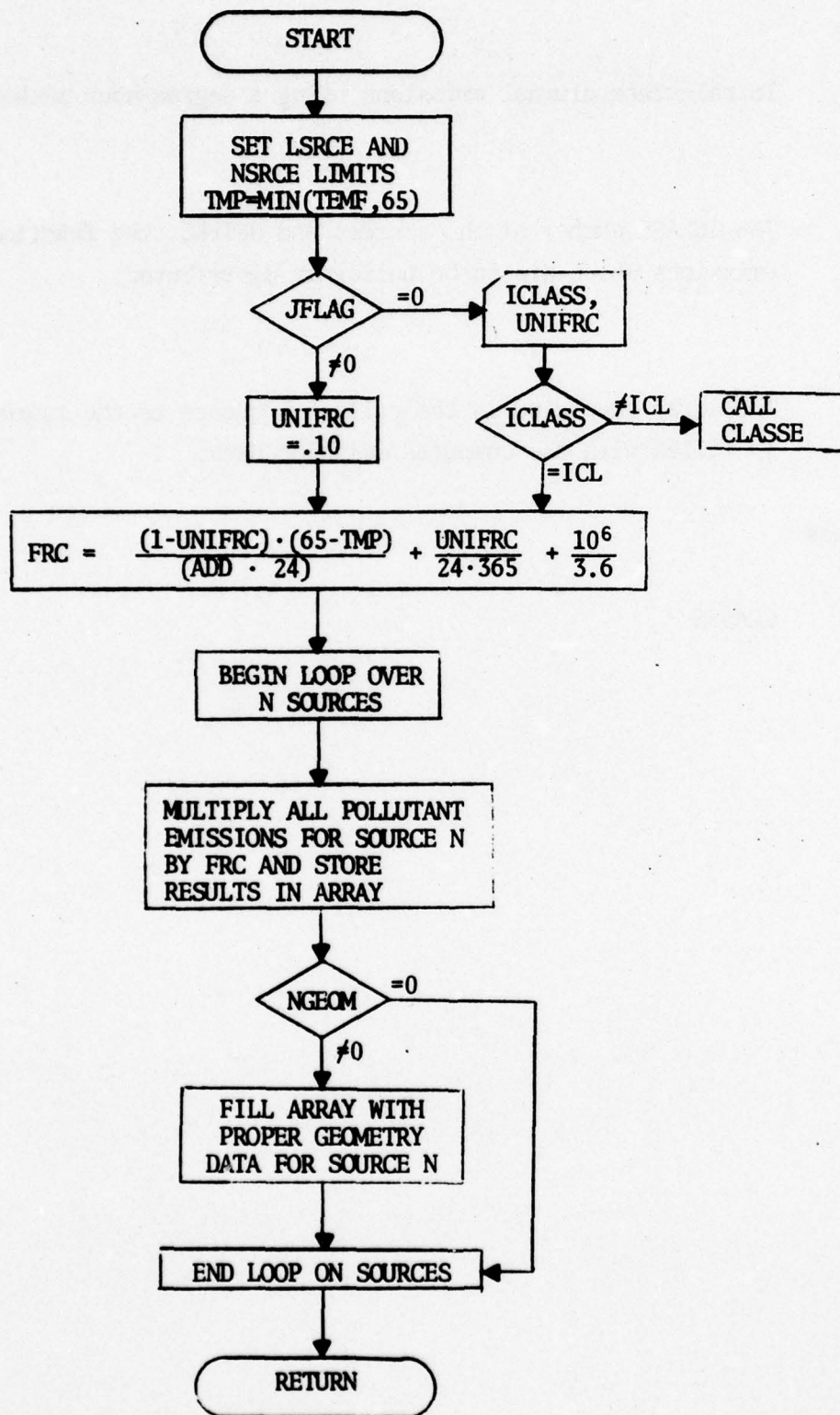
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines Called:

CLASSE



# SUBROUTINE METHB



C	SUBROUTINE METHB(MAXN,ARRAY,I1,I2,ICL)	METHB000
C	THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING	METHB001
C	A DEGREE-HOUR METHOD	METHB002
C		METHB003
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG	METHB004
	COMMON /SPCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,	METHB005
	. NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),	METHB006
	. ABPT(16,150),ABAR(11,100),ABLN(14,100)	METHB007
	COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGH(10,200)	METHB008
	. ,LOC1,LOC2,NGEOM,IPT	METHB009
	COMMON/MET/WS,WSMPH,IWS,WD,IWD,SINWD,COSWD,	METHB010
	. JSTAB,HLID,TEMP,TEMK	METHB011
	DIMENSION AFRAY(I1,I2)	METHB012
	COMMON /ANNMET/ TBAR,ADD,P,PA,WSBAR,DTBAR	METHB013
	LSRCE=NSRCE+1	METHB014
	NSRCE=NSRCE+MAXN	METHB015
	TMP=TEMP	METHB016
	IF (TEMP.GT.65.) TMP=65.	METHB017
	IF (JFLAG.EQ.0) GO TO 5	METHB018
	UNIFRC=.10	METHB019
	GO TO 6	METHB020
	5 READ 1,ICLASS,UNIFRC	METHB021
	1 FCRMAT(I4,4X,F8.7)	METHB022
	IF(ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)	METHB023
	6 FRC=((1.0-UNIFRC)*((65.0-TMP)/(ADD*24.0)))+(UNIFRC/(24.0*365.0))	METHB024
	. * (1.0E+6/3.6)	METHB025
C		METHB026
	DO 30 N=LSRCE,NSRCE	METHB027
	DO 10 I=1,NPLTS	METHB028
	ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC	METHB029
10	CONTINUE	METHB030
	IF (NGEOM.EQ.0) GO TO 30	METHB031
	DO 20 I=1,NGEOM	METHB032
	ARRAY(I,N)=SORCE(I+2,N)	METHB033
20	CONTINUE	METHB034
30	CONTINUE	METHB035
	RETURN	METHB036
	END	METHB037
		METHB038

## SUBROUTINE METHC

### Purpose:

To calculate diurnal emissions using the same distribution pattern for all sources in the class.

### Input:

The ICLASS number of the sources and the fractions of the hour, day and month, FH, FD and FM, which the sources are on. If one or two of the fractions are left blank, the default is used. If all are blank, the sources are assumed to be off.

### Output:

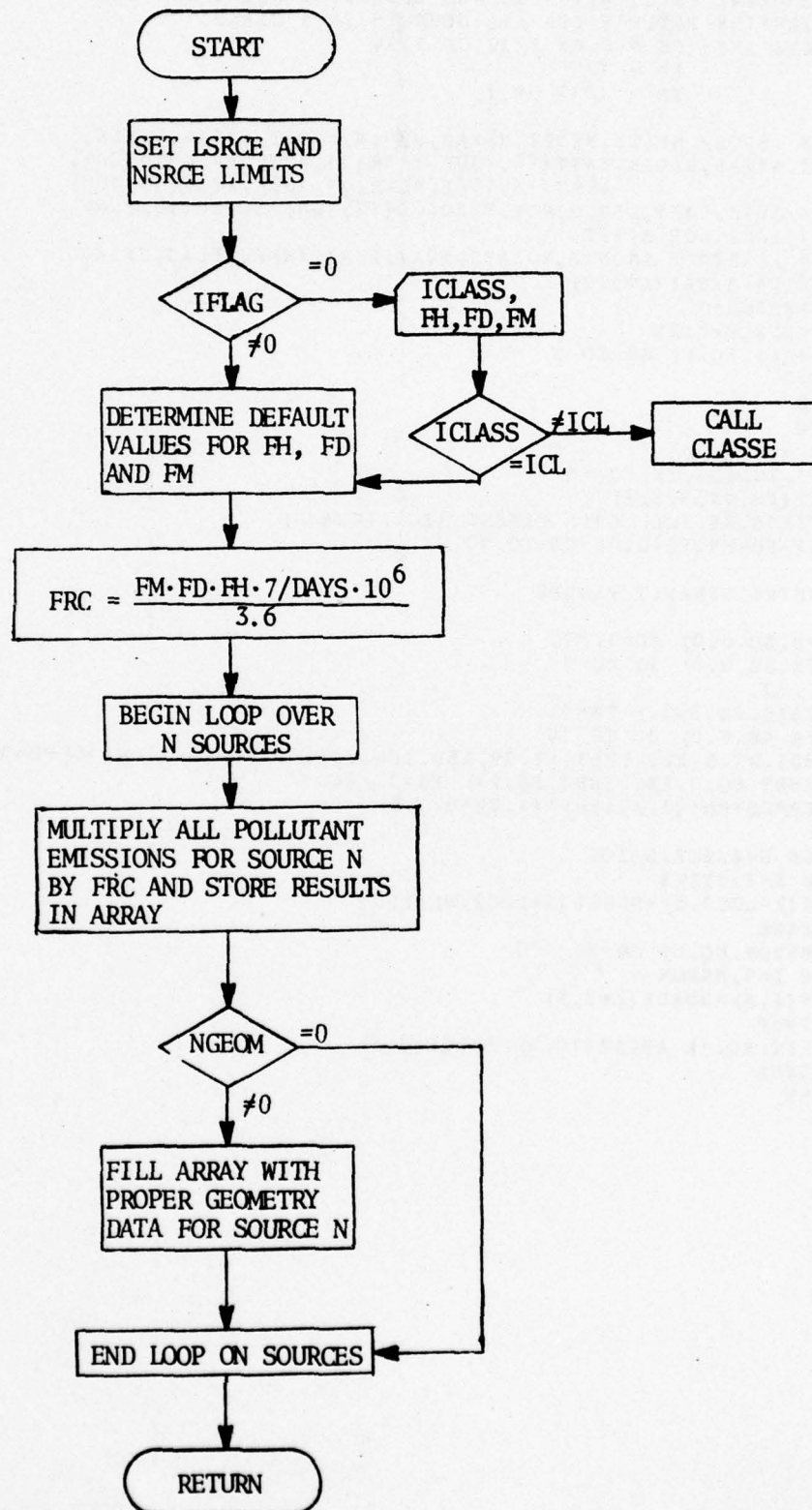
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines Called:

CLASSE



# SUBROUTINE METHC



C	SUBRCUTINE METHC(MAXN,ARRAY,I1,I2,ICL)	METHC000
C	THIS RCUTINE CALCULATES DIURNAL EMISSIONS USING THE SAME	METHC001
C	DISTRIBUTION PATTERN FOR ALL SOURCES IN A CLASS.	METHC002
C	DEFAULTS ARE: FH = 0 CR 1/12 OR 1/24	METHC003
C	FD = 1/7	METHC004
C	FM = 1/12 OR 1	METHC005
C		METHC006
C	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,	METHC007
	. NACPT,NACAR,NACIN,ENET(16,100),ENAR(11,100),ENLN(14,20),	METHC008
	. ABPT(16,150),ABAR(11,100),ABLN(14,100)	METHC009
	CCHMCN/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGH(10,200)	METHC010
	. ,LOC1,LOC2,NGECM,IPT	METHC011
	COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2,IFLAG,JFLAG	METHC012
	DIMENSION ARRAY(I1,I2)	METHC013
	LSRCE=NSRCE+1	METHC014
	NSRCE=NSRCE+MAXN	METHC015
	IF (JFLAG.EQ.0) GO TO 5	METHC016
	FD=1./7.	METHC017
	FM=0.0	METHC018
	FH=0.0	METHC019
	GC TC 6	METHC020
	5 READ 1,ICLASS,FH,FD,FM	METHC021
	1 FORMAT(I4,4X,3F8.7)	METHC022
	IF (ICLASS.NE.ICL) CALL CLASSE (ICL,ICLASS)	METHC023
	IF (FH+FD+FM.EQ.0.0) GO TO 10	METHC024
C		METHC025
C	DETERMINE DEFAULT VALUES	METHC026
C	IF (FD.EQ.0.0) FD=1./7.	METHC027
	6 IF (FM.NE.0.0) GO TO 7	METHC028
	FM=1./12.	METHC029
	IF (DAYS.GE.365.) FM=1.	METHC030
	7 IF (FH.NE.0.0) GC TO 10	METHC031
	IF (IHR1.GT.6.AND.IHR1.LT.19.AND.IHR2.GT.6.AND.IHR2.LT.19) FH=1./12.	METHC032
	IF (IHR1.EQ.1.AND.IHR2.EQ.24) FH=1./24.	METHC033
	10 FRC=FM*FD*FH*(7./DAYS)*(1.0E+6/3.6)	METHC034
C		METHC035
	20 DC 100 N=LSRCE,NSRCE	METHC036
	DO 30 I=1,NPLTS	METHC037
	ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC	METHC038
	30 CCNTINUE	METHC039
	IF (NGEOM.EQ.0) GO TO 100	METHC040
	DO 40 I=1,NGEOM	METHC041
	ARRAY(I,N)=SORCE(I+2,N)	METHC042
	40 CCNTINUE	METHC043
	IF (IPT.EQ.1) ARRAY(10,N)=SORCE(2,N)	METHC044
	100 CONTINUE	METHC045
	RETURN	METHC046
	END	METHC047
		METHC048
		METHC049

# SUBROUTINE METHD

## Purpose:

To calculate diurnal emissions using the temporal distribution arrays for fuel handling activities.

## Input:

None

## Output:

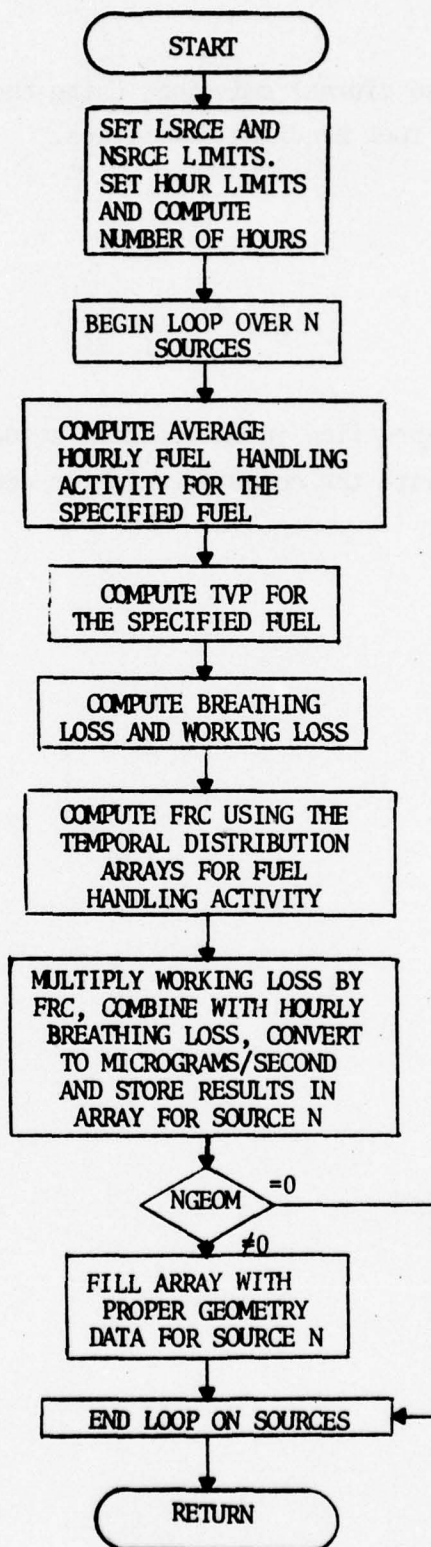
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

## Subroutines Called:

None



SUBROUTINE METHD



	SUBROUTINE METHD (MAXN, ARRAY, I1, I2)	METHD000
C		METHD001
C	THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE	METHD002
C	TEMPORAL DISTRIBUTION ARRAYS FOR FUEL HANDLING ACTIVITIES	METHD003
C		METHD004
	COMMON /SRCE/ NPLTS, NENPT, NENAR, NENLN, NABPT, NABAR, NABLN,	METHD005
	. NACPT, NACAR, NACLN, ENPT (16, 100), ENAR (11, 100), ENLN (14, 20),	METHD006
	. ABPT (16, 150), ABAR (11, 100), ABLN (14, 100)	METHD007
	COMMON/JUNK/DAYS, LSRCE, NSPCE, SORCE (17, 300), SORGM (10, 200)	METHD008
	. , LOC1, LOC2, NGEOM, IFT	METHD009
	COMMON/EEPIOD/IMONTH, NODAYS, IDAY, IHR1, IHR2	METHD010
	COMMON/MET/WS, WSMPL, IWS, WD, IWD, SINWD, COSWD,	METHD011
	. JSTAE, HLID, TEMP, TEMK	METHD012
	COMMON / DEFAULT / ITAPE, ACLNDY, ACLNDZ, ALPHA (7), BETA (7), FLDENS (7)	METHD013
	COMMON /DSTRET/ ACHO (13, 8), ACDY (2, 8), ACHR (24, 8), VHMLMO (13),	METHD014
	. VHMLDY (2), VHMLHR (24), CVABMO (13), CVABDY (2), CVABHR (24), CVENMO (13),	METHD015
	. CVENDY (2), CVENHR (24), FLMO (13, 7), FLDY (2, 7), FLHR (24, 7), NC1	METHD016
	COMMON/MONMET/ TMBAR	METHD017
	DIMENSION ARRAY (I1, I2)	METHD018
	LSRCE=NSRCE+1	METHD019
	NSRCE=NSRCE+MAXN	METHD020
	NHI=IHR2	METHD021
	IF (IHR1.GT.IHR2) NHI=24+IHR2	METHD022
	HRS=NHI-IHR1 + 1	METHD023
	DC 30 N=LSRCE, NSRCE	METHD024
	FLHOUR=0.	METHD025
	IDF=SORCE (14, N)	METHD026
	DC 10 I=IHR1, NHI	METHD027
	II=I	METHD028
	IF (I.GT.24) II=I-24	METHD029
	FLHOUR=FLHR (II, IDF) + FLHOUR	METHD030
10	CCONTINUE	METHD031
	FLHOUR=FLHOUR/HRS	METHD032
	TVP=EXP (ALPHA (IDF) - BETA (IDF) / (5. * (TMBAR-32.) / 9. + 273.))	METHD033
	BRLOSS=SORCE (13, N) * (TVP / (14.7 - TVP)) ** 0.69	METHD034
	WRKLOS=SORCE (12, N) * TVP	METHD035
	FRC=FLMO (IMONTH, IDF) * FLDY (IDAY, IDF) * FLHOUR * (7./DAYS)	METHD036
		METHD037
C	ARRAY (12, N) = (BRLOSS / (365. * 24.) + WRKLOS * FRC) * 1.E+6 / 3.6	METHD038
	IF (IFT.EQ.1) ARRAY (10, N) = SORCE (2, N)	METHD039
	IF (NGEOM.EQ.0) GO TO 30	METHD040
	DC 20 I=1, NGEOM	METHD041
	ARRAY (I, N) = SORCE (I+2, N)	METHD042
20	CONTINUE	METHD043
30	CCONTINUE	METHD044
	RETURN	METHD045
	END	METHD046

## SUBROUTINE METHE

### Purpose:

To calculate diurnal emissions using the temporal distribution arrays for vehicle activities.

### Input:

None

### Output:

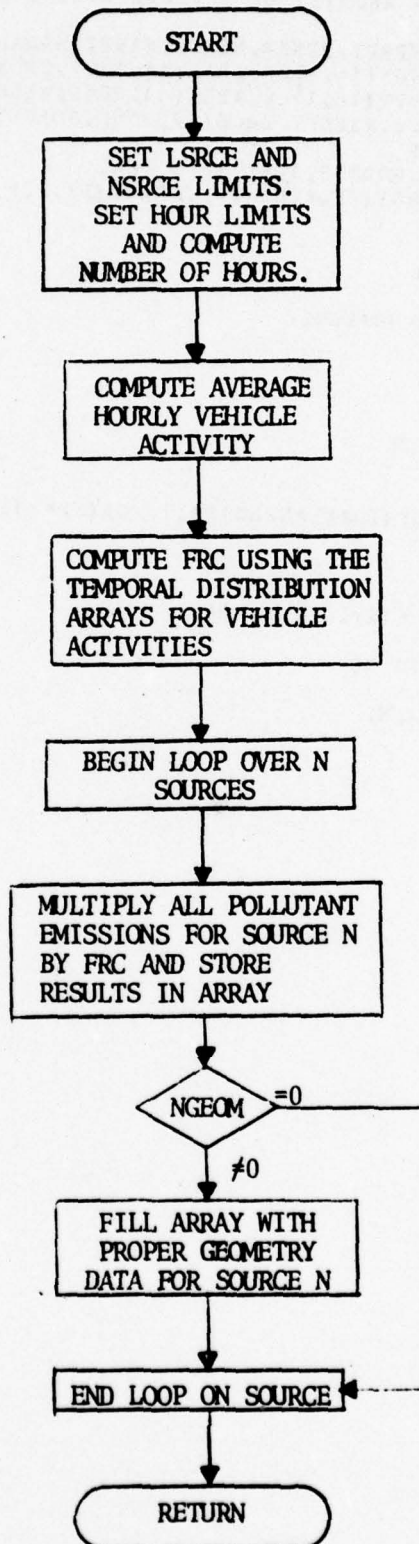
The array specified in the calling sequence to the subroutine is filled with the computed emission data.

### Subroutines Called:

None



SUBROUTINE METHE



C	SUBROUTINE METHE(MAXN,ARRAY,ARMO,ARDY,ARHR,I1,I2)	METHE000
C	THIS ROUTINE CALCULATES DIURNAL EMISSIONS USING THE	METHE001
C	TEMPORAL DISTRIBUTION ARRAYS FOR VEHICLE ACTIVITIES	METHE002
C		METHE003
	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,	METHE004
	. NACPT,NACAR,NACLN,ENPT(16,100),ENAR(11,100),ENLN(14,20),	METHE005
	. ABPT(16,150),ABAR(11,100),ABLN(14,100)	METHE006
	COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)	METHE007
	. ,LOC1,LOC2,NGEOM,IPT	METHE008
	COMMON/PERIOD/IMONTH,NODAYS,IDAY,IHR1,IHR2	METHE009
	DIMENSION ARMO(13),ARDY(2),ARHR(24),ARRAY(11,12)	METHE010
	LSRCE=NSRCE+1	METHE011
	NSRCE=NSRCE+MAXN	METHE012
	AFHCUF=0.	METHE013
	NHI=IHR2	METHE014
	IF(IHR1.GT.IHR2) NHI=24+IHR2	METHE015
	HRS=NHI-IHR1+1	METHE016
	DO 10 I=IHR1,NHI	METHE017
	II=I	METHE018
	IF(I.GT.24) II=I-24	METHE019
	ARHCUF=ARHOUR+ARHR(II)	METHE020
10	CONTINUE	METHE021
	ARHOUR=ARHOUR/HRS	METHE022
	FRC=ARMO(IMONTH)*ARDY(IDAY)*ARHOUR*(7./DAYS)*(1E+6/3.6)	METHE023
C		METHE024
	DO 40 N=LSRCE,NSRCE	METHE025
	DO 20 I=1,NPLTS	METHE026
	ARRAY(I+LOC1,N)=SORCE(I+LOC2,N)*FRC	METHE027
20	CONTINUE	METHE028
	IF(NGEOM.EQ.C) GO TO 40	METHE029
	DO 30 I=1,NGEOM	METHE030
30	ARRAY(I,N)=SORCE(I+2,N)	METHE031
40	CONTINUE	METHE032
	RETURN	METHE033
	END	METHE034
		METHE035

## SUBROUTINE OUTPUT

### Purpose:

To print the pollutant concentrations at all receptors for the environ, airbase, aircraft and total combined sources.

### Input:

1. Title information.
2. The RECEP and RECDAT arrays containing receptor and concentration data.

### Output:

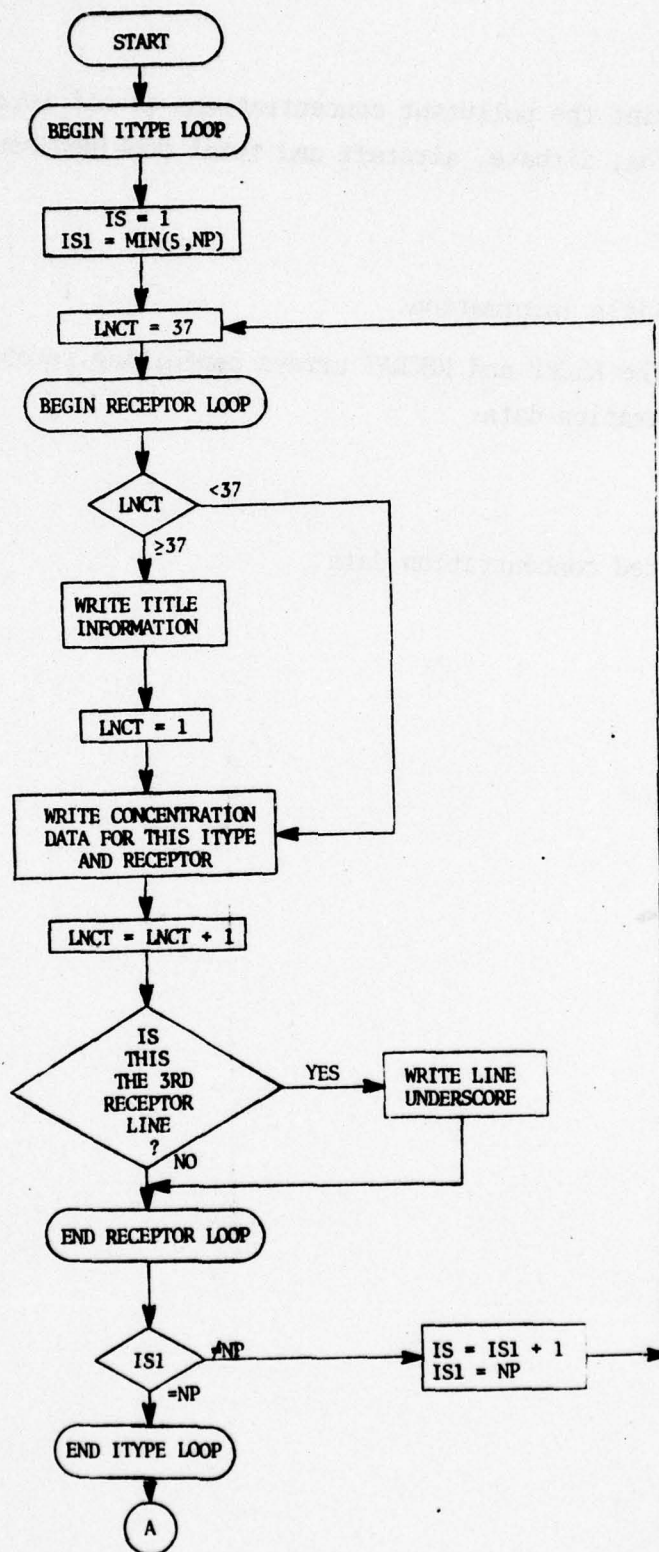
Printed concentration data.

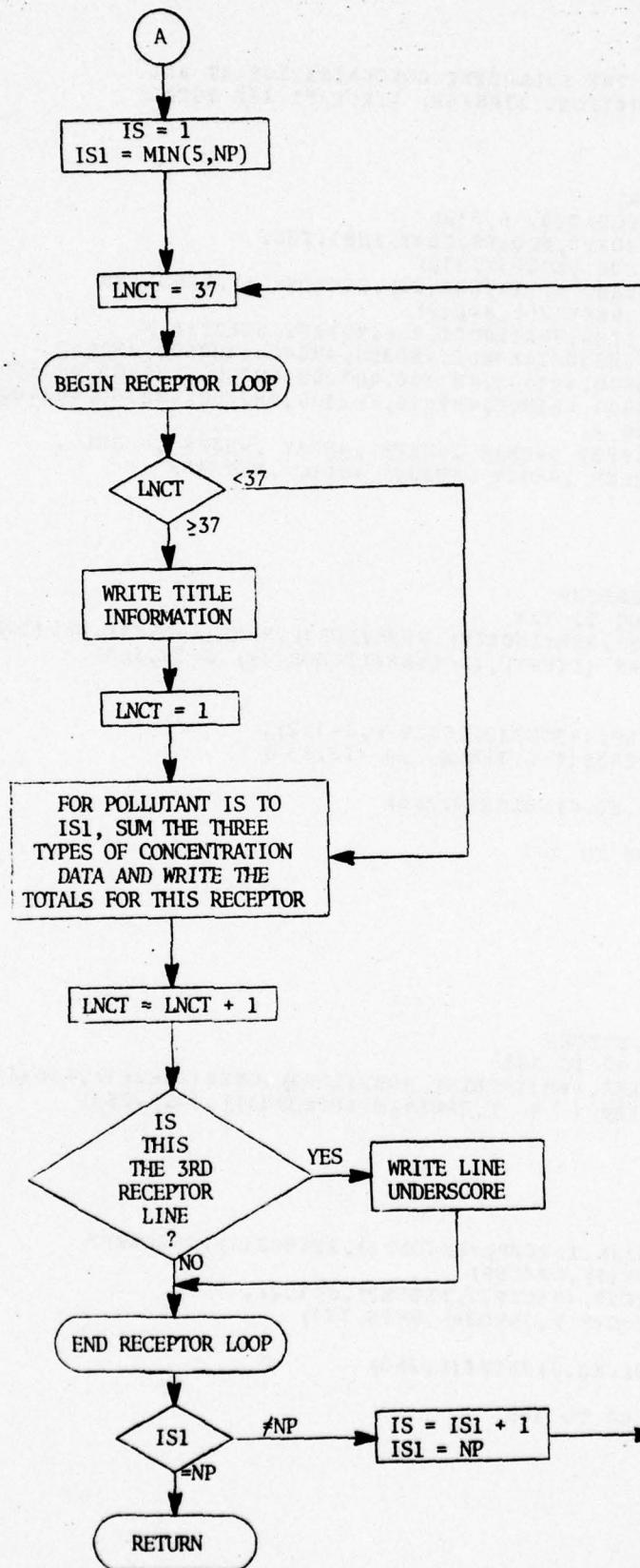
### Subroutines Called:

None



# SUBROUTINE OUTPUT





# SUBROUTINE OUTPUT

THIS ROUTINE PRINTS THE POLLUTANT CONCENTRATION AT ALL RECEPTORS FOR THE ENVIRON, AIRBASE, AIRCRAFT AND TOTAL COMBINED SOURCES.

```

REAL*8 POLNAM
REAL*8 SORNAM(4)
COMMON /AIRQAL/ RECDAT(3, 6,312)
COMMON /PERIOD/ IMONTH,NODAYS,IDAY,IHR1,IHR2
COMMON /RCPT/ NRECEP,RECEP(2,312)
COMMON /TITL/ POLNAM( 6),TITLE1(20),IPCHOS( 6),NXPOL,NP
DIMENSION NNH(13),NNHR(25),NND(2)
DATA SORNAM/7HENVIRON,7HAIRPORT,8HAIRCRAFT,5HTOTAL /
DATA NNHR/4H0000,4H0100,4H0200,4H0300,4H0400,4H0500,4H0600,
.4H0700,4H0800,4H0900,4H1000,4H1100,4H1200,4H1300,4H1400,4H1500,
.4H1600,4H1700,4H1800,4H1900,4H2000,4H2100,4H2200,4H2300,4H2400/,
. NND /4HDAY ,4HEND /,
. NNH/4HJAN ,4HFEB ,4HMAR ,4HAPR ,4HMAY ,4HJUN ,4HJUL ,
. 4HAUG ,4HSEP ,4HOCT ,4HNOV ,4HDEC ,4HYEAR/
DO 100 ITYPE=1,3
IS=1
IS1=MINO(5,NP)
110 LNCT=37
DO 120 IRECEP=1,NRECEP
IF (LNCT.LT.37) GO TO 121
WRITE (6,220) TITLE1,NNH(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
WRITE (6,200) SORNAM (ITYPE),(POLNAM(IPCHOS(J)),J=IS,IS1)
WRITE (6,260)
LNCT=1
121 WRITE (6,210) IRECEP,(RECEP(J,IRECEP),J=1,2),
.(RECDAT(ITYPE,IPCHOS(K),IRECEP),K=IS,IS1)
LNCT=LNCT+1
IF (MOD(IRECEP,3).EQ.0) WRITE(6,260)
120 CCNTINUE
IF (IS1.EQ.NP) GO TO 100
IS=IS1+1
IS1=NP
GC TO 110
100 CONTINUE
IS=1
IS1=MINO(5,NP)
125 LNCT=37
DO 130 IRECEP=1,NRECEP
IF (LNCT.LT.37) GO TO 133
WRITE (6,220) TITLE1,NNH(IMONTH),NNHR(IHR1),NNHR(IHR2+1),NND(IDAY)
WRITE (6,200) SORNAM ( 4 ),(POLNAM(IPCHOS(J)),J=IS,IS1)
WRITE (6,260)
LNCT=1
133 CCNTINUE
DO 131 J=IS,IS1
DO 131 K=2,3
131 RECDAT(1,IPCHOS(J),IRECEP)=RECDAT(1,IPCHOS(J),IRECEP) +
. RECDAT(K,IPCHOS(J),IRECEP)
WRITE (6,210) IRECEP,(RECEP(J,IRECEP),J=1,2),
.(RECDAT(1,IPCHOS(K),IRECEP),K=IS,IS1)
LNCT=LNCT+1
IF (MOD(IRECEP,3).EQ.0) WRITE(6,260)
130 CONTINUE
IF (IS1.EQ.NP) GO TO 140
IS=IS1+1
IS1=NP

```

OUTPT000  
 OUTPT001  
 OUTPT002  
 OUTPT003  
 OUTPT004  
 OUTPT005  
 OUTPT006  
 OUTPT007  
 OUTPT008  
 OUTPT009  
 OUTPT010  
 OUTPT011  
 OUTPT012  
 OUTPT013  
 OUTPT014  
 OUTPT015  
 OUTPT016  
 OUTPT017  
 OUTPT018  
 OUTPT019  
 OUTPT020  
 OUTPT021  
 OUTPT022  
 OUTPT023  
 OUTPT024  
 OUTPT025  
 OUTPT026  
 OUTPT027  
 OUTPT028  
 OUTPT029  
 OUTPT030  
 OUTPT031  
 OUTPT032  
 OUTPT033  
 OUTPT034  
 OUTPT035  
 OUTPT036  
 OUTPT037  
 OUTPT038  
 OUTPT039  
 OUTPT040  
 OUTPT041  
 OUTPT042  
 OUTPT043  
 OUTPT044  
 OUTPT045  
 OUTPT046  
 OUTPT047  
 OUTPT048  
 OUTPT049  
 OUTPT050  
 OUTPT051  
 OUTPT052  
 OUTPT053  
 OUTPT054  
 OUTPT055  
 OUTPT056  
 OUTPT057  
 OUTPT058  
 OUTPT059  
 OUTPT060  
 OUTPT061



GC TO 125	OUTPT062
140 CCNTINUE	OUTPT063
200 FORMAT(1H0,96(1H-)/2H I,22X,33HRECEPTOR CONCENTRATION DATA FROM ,	OUTPT064
. A8,8H SOURCES,23X,1HI/2H I,94(1H-),1HI/	OUTPT065
. 37H I RECEPTOR I RECEPTOR LOCATION I,17X,24HEXPECTED ARITHMET	OUTPT066
.IC MEAN,18X,1HI/13H I NUMBER I,23X,1HI,59X,1HI/	OUTPT067
. 2H I,10(1H-),1HI,23(1H-),1HI,59(1H-),1HI/	OUTPT068
. 2H I,10X,1HI,5X,12H(KILOMETERS),6X,1HI,18X,22H(MICROGRAMS/CU. MET	OUTPT069
.ER),19X,1HI,/2H I,10X,1HI,5X,1HX,5X,1HI,5X,1HY,	OUTPT070
. 5X,4(3HI ,A8,1X),3HI ,A8,2H I)	OUTPT071
210 FCRMAT(2H I,I6,4X,2(1HI,F9.3,2X),1HI,5(1PE10.3,2H I))	OUTPT072
220 FCRMAT(1H1,9X,20A4/10H MONTH = ,A4,12H PERIOD = ,A4,4H TO ,	OUTPT073
. A4,16H HOURS ON A WEEK,A4)	OUTPT074
260 FORMAT(2HI,10(1H-),1HI,7(11(1H-),1HI))	OUTPT075
RETURN	OUTPT076
END	OUTPT077

## SUBROUTINE PLRISE

### Purpose:

To calculate the effective height and the vertical and horizontal dispersion coefficients for a given stack.

### Input:

The stack parameters and current meteorological conditions.

### Output:

1. The effective height,  $h_{eff}$ .
2. The vertical and horizontal dispersion coefficients,  $\sigma_{yo}$  and  $\sigma_{zo}$ .
3. KSTAB, a flag used in the TRAN function
  - = 0, the modified stack height is below the lid
  - = 1, the modified stack height is initially above the lid
  - = 2, the plume will penetrate the lid.

### Procedure:

1. For point sources having no plume rise:
$$h_{eff} = \max (Z_S, H_B, \Delta Z/2.)$$
$$\sigma_{yo} = \Delta Y/2.4$$
$$\sigma_{zo} = \Delta Z/2.4$$
$$KSTAB = 0 \text{ or } 1$$
2. For point sources which may undergo plume rise:
  - a. Estimate the wind speed at the top of the aerovane
  - b. Modify the stack height by the effect of the stack downwash
  - c. Test for building downwash effects. If downwash occurs:
$$h_{eff} = H_B + .5L_B$$
$$\sigma_{yo} = \sigma_{zo} = h_{eff}/1.2$$
$$KSTAB = 0 \text{ or } 1$$
  - d. Test to determine if the buoyant plume rise is significant.

e. Check for an inversion

f. Compute the plume rise using function RISE

g. If no downwash occurs:

$$H_{\text{eff}} = Z_S + 2 \left( \frac{VS}{U_a} - 1.5 \right) \cdot DS + \text{plume rise}$$

$$\sigma_{yo} = \Delta Y / 2.4$$

$$\sigma_{zo} = \Delta Z / 2.4$$

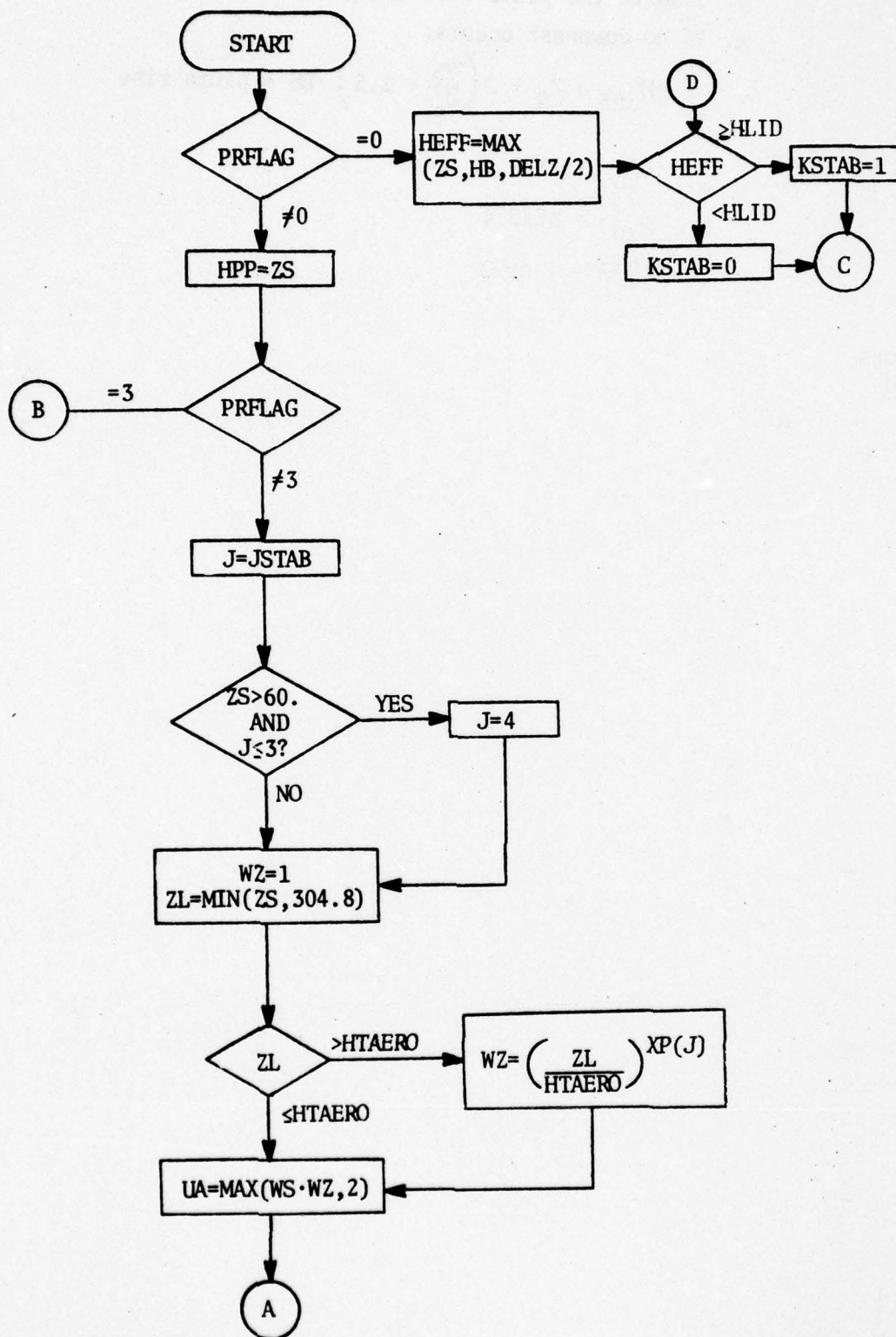
$$KSTAB = 1 \text{ or } 2$$

Functions  
Called:

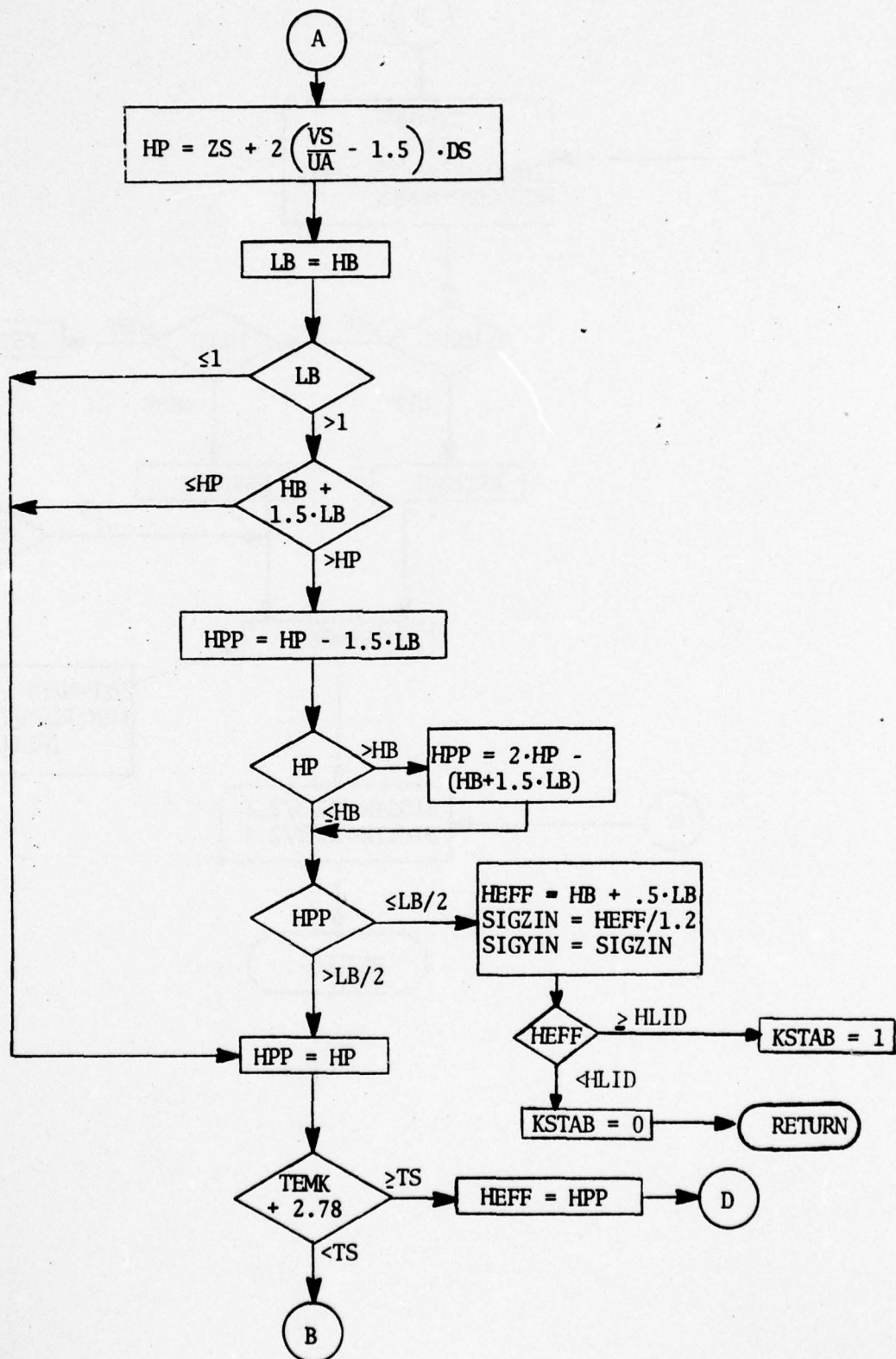
RISE



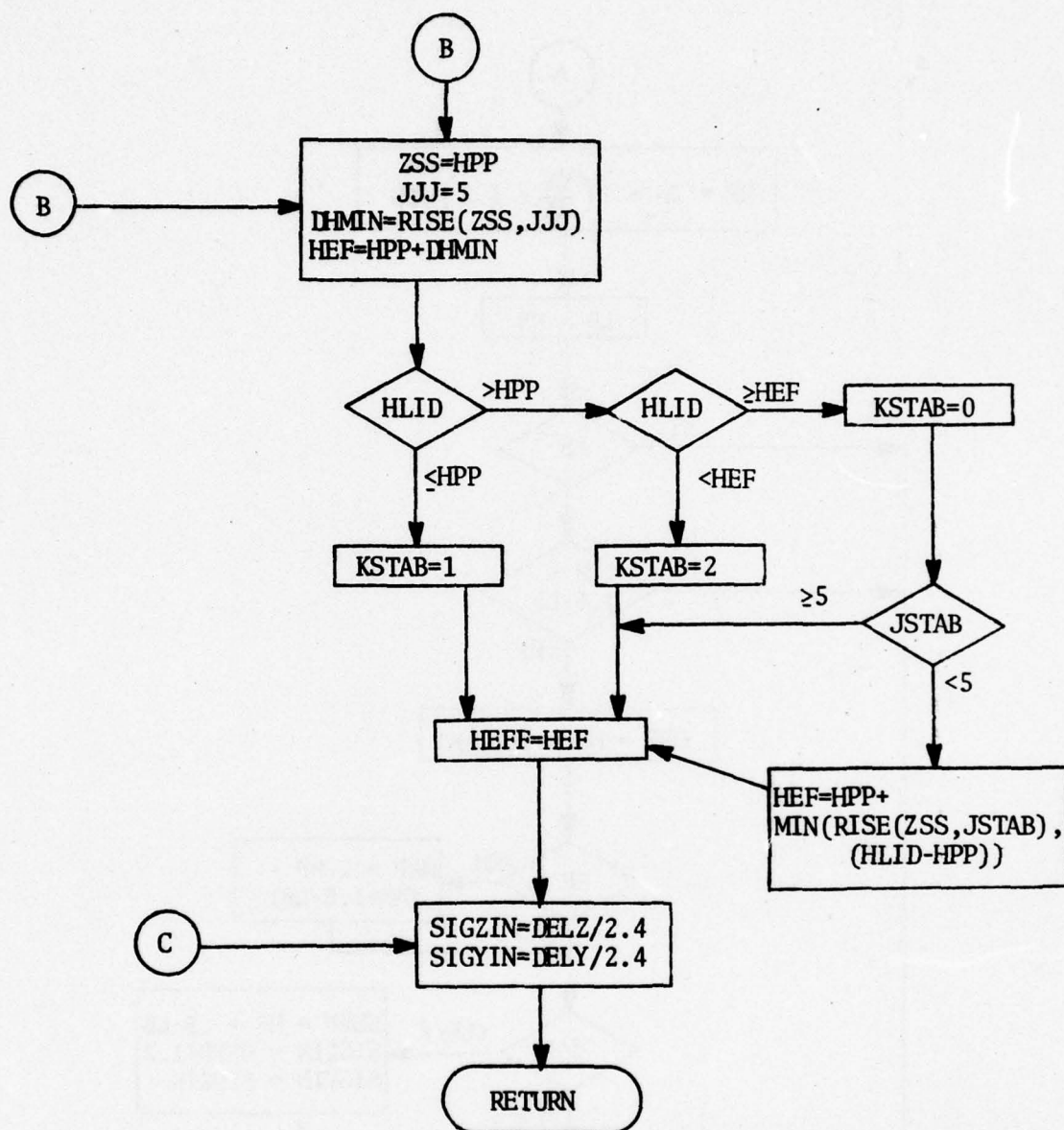
# SUBROUTINE PLRISE



SUBROUTINE PLRISE (Cont'd.)



SUBROUTINE PLRISE (Cont'd.)





```

      SUBROUTINE PLR1SE(HEFF,KSTAB,SIGZIN,SIGYIN)
C
C   THIS SUBROUTINE CALCULATES THE EFFECTIVE HEIGHT AND THE
C   VERTICAL AND HORIZONTAL DISPERSION COEFFICIENTS
C   FOR A GIVEN STACK
C
      REAL LP
      COMMON /MET/ VS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,
      . TEMK,UA
      COMMON /INFO/ IRECEP,IWDIR,ITYPE,HTAFRO,XS,YS,ZS,DELY,DELZ,
      . TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL
      COMMON /WINDPRO/ XP(6)
      IF (PRFLAG.NE.0) GO TO 100
C
C   FOR AN AREA SOURCE WITH A DIAMETER OF LESS THAN 50 METERS
C   THE EFFECTIVE EMISSION HEIGHT IS SET TO THE MAXIMUM OF
C   Z, THE BUILDING HEIGHT OR DELTA Z/2.0
C
      HEFF=AMAX1(ZS,HB,DELZ/2.)
      50 KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      GO TO 230
C
      100 CONTINUE
      HFP=ZS
      IF (PRFLAG.EQ.3) GO TO 130
C
C   FIRST TEST FOR DOWNWASH, THEN COMPUTE PLUME RISE, IF ANY
C
C   FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW
C
      J=JSTAB
      IF (ZS.GT.60..AND.J.LE.3) J=4
C
C   COMPUTE THE WINDSPEED AT THE ELEVATION OF THE STACK
C
      WZ=1.0
      ZL=AMIN1(ZS,304.8)
      IF (ZL.GT.HTAFRO) WZ=(ZL/HTAFRO)**XP(J)
      UA=AMAX1(VS*WZ,2.0)
C
C   COMPUTE STACK DOWNWASH
C
      HF=ZS+2.0*(VS/UA-1.5)*DS
      LP=HF
C
C   BUILDING DOWNWASH TESTS
C
      IF (LB.LE.1.) GO TO 110
      IF (HF.GE.(HB+1.5*LB)) GO TO 110
      HPP=HF-1.5*LB
      IF (HP.GT.HB) HPP=2.0*HP-(HB+1.5*LB)
      IF (HPP.GT.(LP/2.0)) GO TO 110
C
C   BUILDING DOWNWASH OCCURS
C
      HEFF=HF+0.5*LP
      SIGZIN=HEFF/1.2
      SIGYIN=SIGZIN
      KSTAB=0
      IF (HEFF.GE.HLID) KSTAB=1
      RETURN

```

```

PLRSE000
PLRSE001
PLRSE002
PLRSE003
PLRSE004
PLRSE005
PLRSE006
PLRSE007
PLRSE008
PLRSE009
PLRSE010
PLRSE011
PLRSE012
PLRSE013
PLRSE014
PLRSE015
PLRSE016
PLRSE017
PLRSE018
PLRSE019
PLRSE020
PLRSE021
PLRSE022
PLRSE023
PLRSE024
PLRSE025
PLRSE026
PLRSE027
PLRSE028
PLRSE029
PLRSE030
PLRSE031
PLRSE032
PLRSE033
PLRSE034
PLRSE035
PLRSE036
PLRSE037
PLRSE038
PLRSE039
PLRSE040
PLRSE041
PLRSE042
PLRSE043
PLRSE044
PLRSE045
PLRSE046
PLRSE047
PLRSE048
PLRSE049
PLRSE050
PLRSE051
PLRSE052
PLRSE053
PLRSE054
PLRSE055
PLRSE056
PLRSE057
PLRSE058
PLRSE059
PLRSE060
PLRSE061

```

C		PLRSE062
C	NC BUILDING DOWNWASH, TEST FOR PLUME RISE	PLRSE063
C		PLRSE064
	110 HEP=HP	PLRSE065
	IF (TS.GT.(TEMK+2.78)) GO TO 130	PLRSE066
C		PLRSE067
C	COLD PLUME	PLRSE068
C		PLRSE069
	HEFF=HPP	PLRSE070
	GC TC 50	PLRSE071
	130 CONTINUE	PLRSE072
C		PLRSE073
C	PLUME RISE EXPECTED TO BE SIGNIFICANT	PLRSE074
C	CALCULATE MINIMUM PLUME RISE	PLRSE075
C		PLRSE076
	ZSS=HFP	PLRSE077
	JJJ=5	PLRSE078
	DHMIN=RISE(ZSS,JJJ)	PLRSE079
	HEF=HPP+DHMIN	PLRSE080
C		PLRSE081
C	TEST FOR INTERFERENCE OF LID WITH MODIFIED PHYSICAL STACK	PLRSE082
C	HEIGHT AND PLUME	PLRSE083
C		PLRSE084
	IF (HLID.GT.HPP) GO TO 220	PLRSE085
C		PLRSE086
C	LID INTERFERES WITH STACK HEIGHT, USE STABILITY 5 WITH	PLRSE087
C	INFINITE LID HEIGHT	PLRSE088
C		PLRSE089
	KSTAB=1	PLRSE090
	GO TO 225	PLRSE091
C		PLRSE092
C	LID INTERFERES WITH PLUME, USE STABILITY 5 WITH INFINITE LID	PLRSE093
C		PLRSE094
	220 IF (HLID.GE.HEF) GO TO 221	PLRSE095
	KSTAB=2	PLRSE096
	GO TO 225	PLRSE097
C		PLRSE098
C	CALCULATE PLUME RISE, PLUME CANNOT PENETRATE THE LID	PLRSE099
C		PLRSE100
	221 KSTAB=0	PLRSE101
	IF (JSTAB.LT.5) HEF=HPP+AMTN1(RISE(ZSS,JSTAB),(HLID-HPP))	PLRSE102
	225 CONTINUE	PLRSE103
	HEFF=HEF	PLRSE104
	230 SIGZIN=DELZ/2.4	PLRSE105
	SIGYIN=DELY/2.4	PLRSE106
	RETURN	PLRSE107
	END	PLRSE108

## SUBROUTINE POLSOR

### Purpose:

To direct the calls to the proper diffusion routine for all input sources.

### Input:

1. Point source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft
2. Area source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft
3. Line source data for:
  - a. Environs
  - b. Airbase
  - c. Aircraft

### Output:

SORC, a vector which contains data for the current source to be transferred to the diffusion models.

### Procedure:

1. Set the receptor data array to zero.
2. Set the type flag for environs, airbase or aircraft.
3. Fill the SORC vector with the source description and emission parameters.
4. Check for non-zero emissions from this source and call STPOL1 for point and area sources and STPOL2 for line sources.

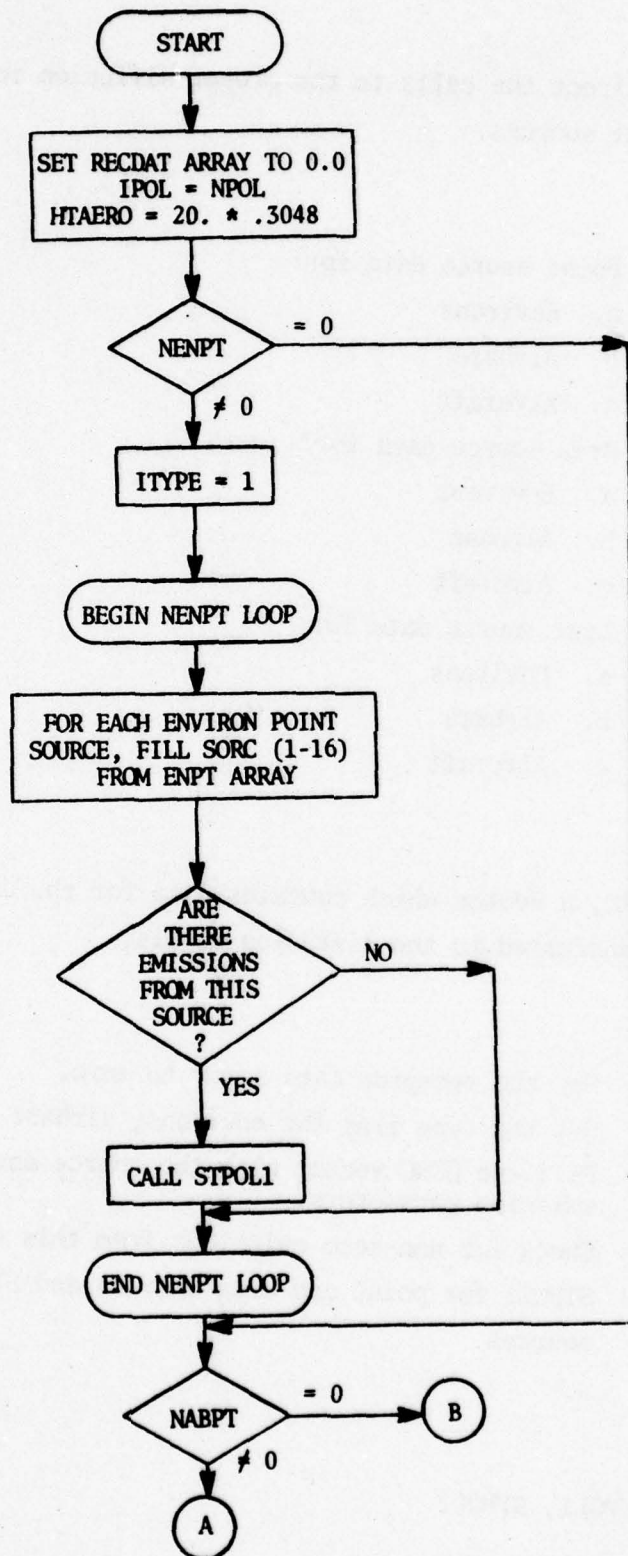
### Subroutines

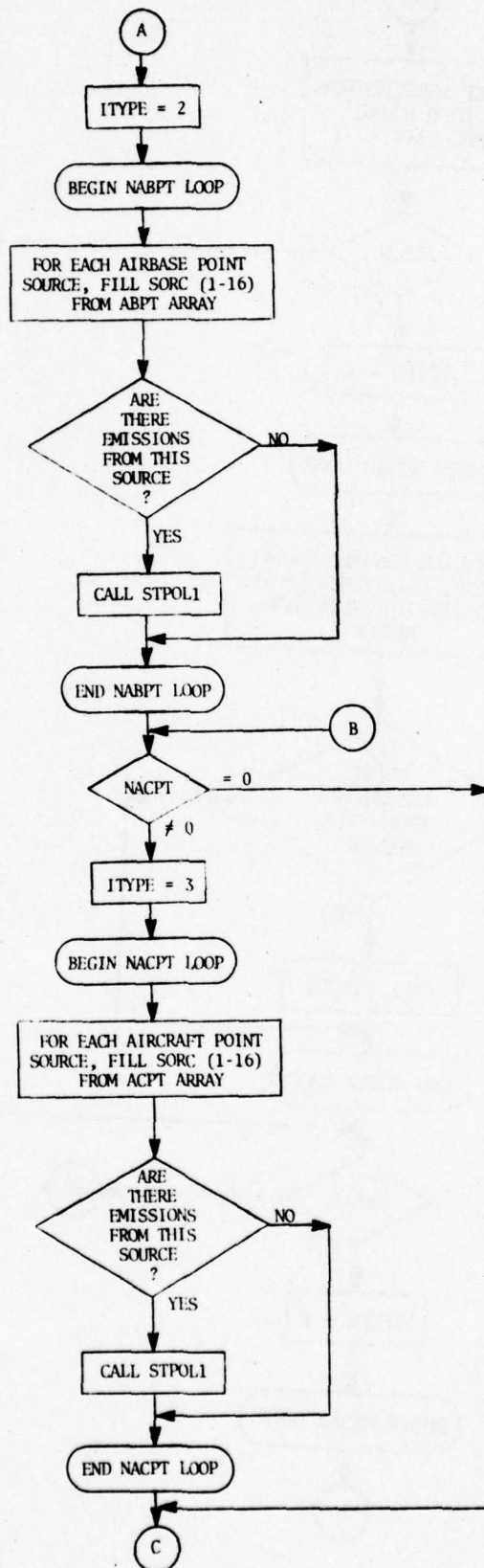
#### Called:

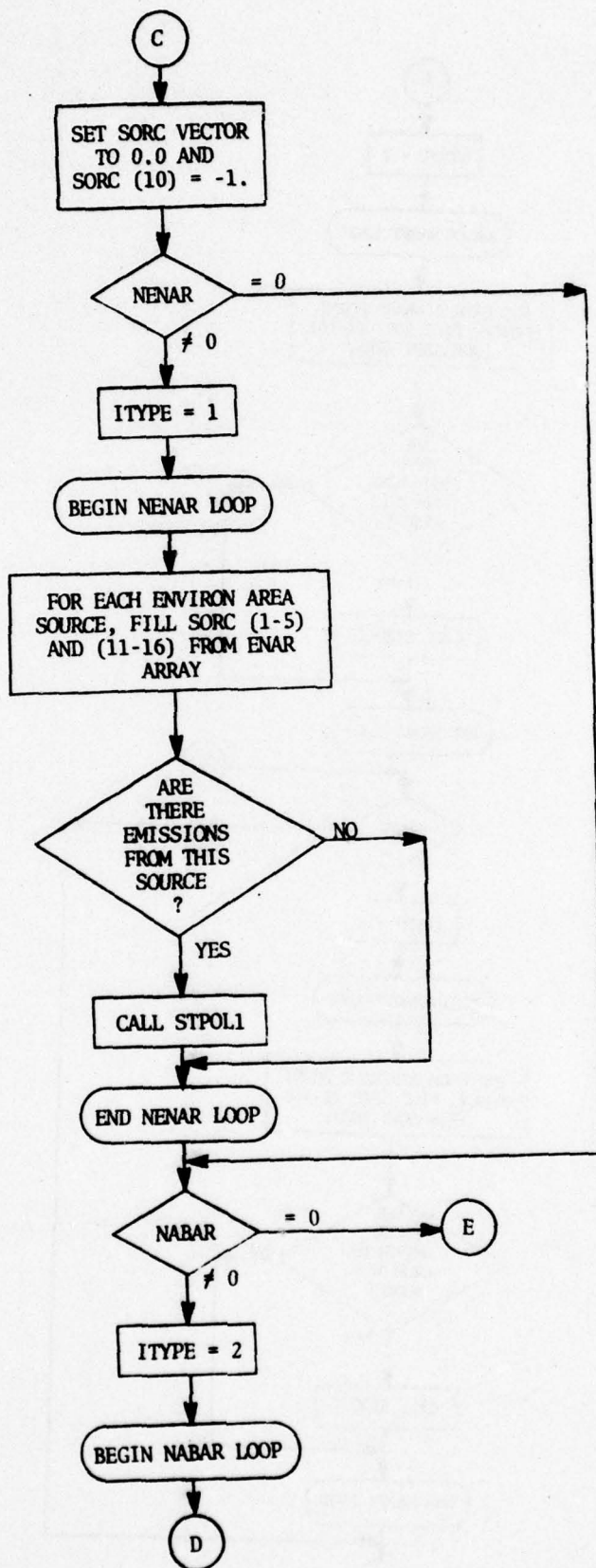
STPOL1, STPOL2



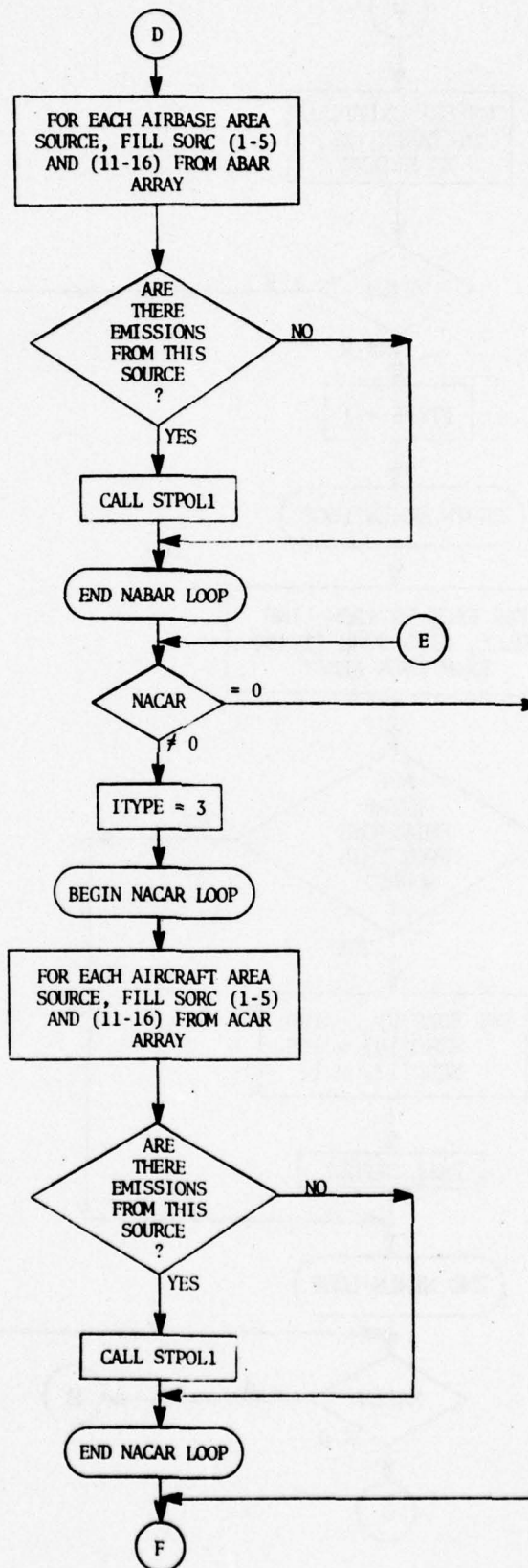
# SUBROUTINE POLSOR

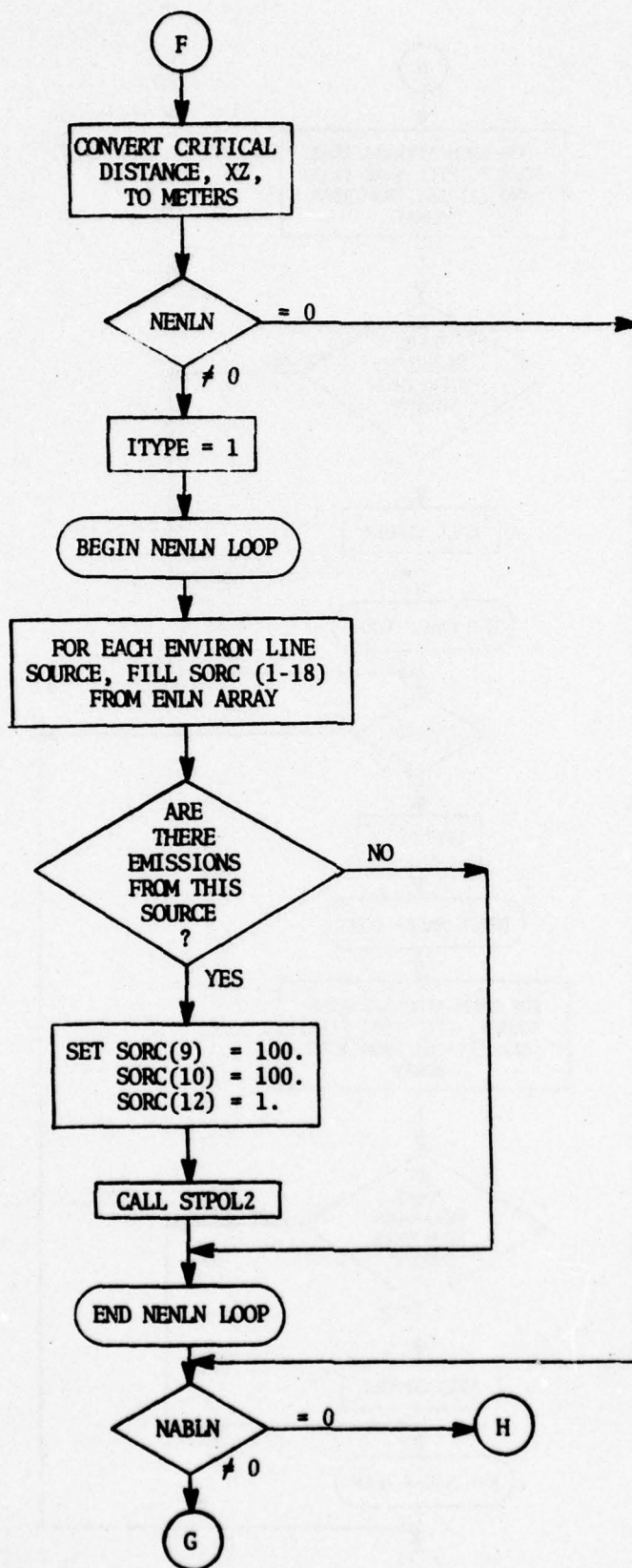


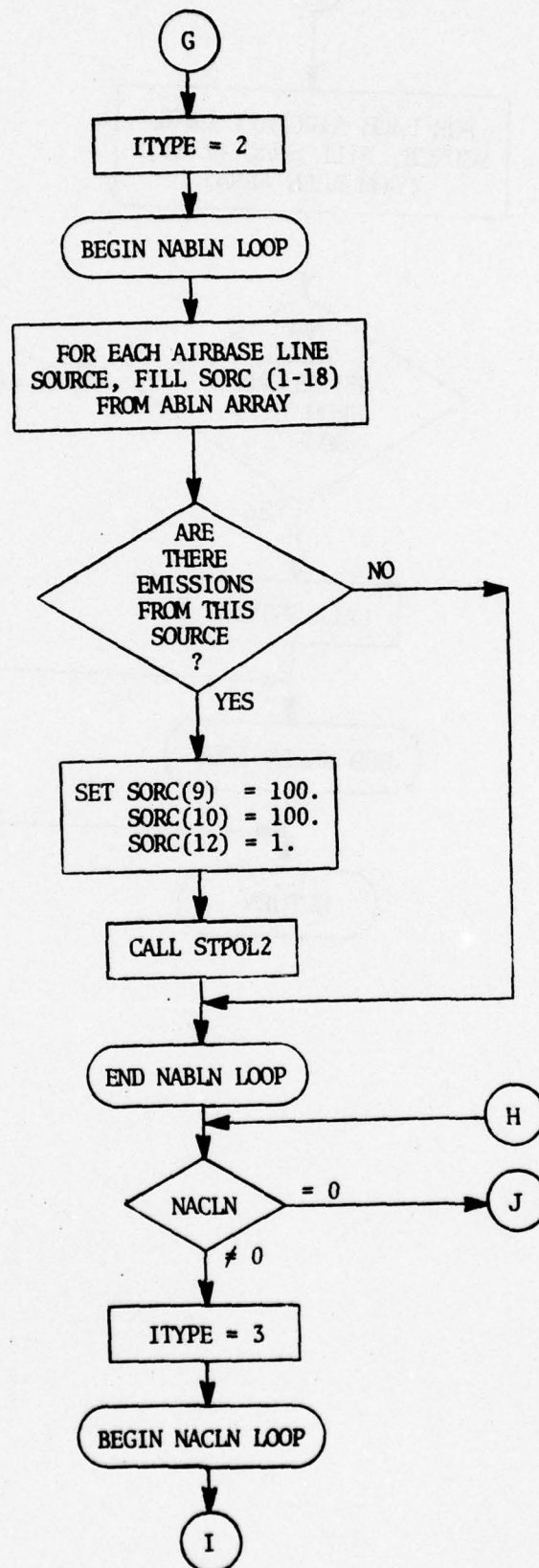




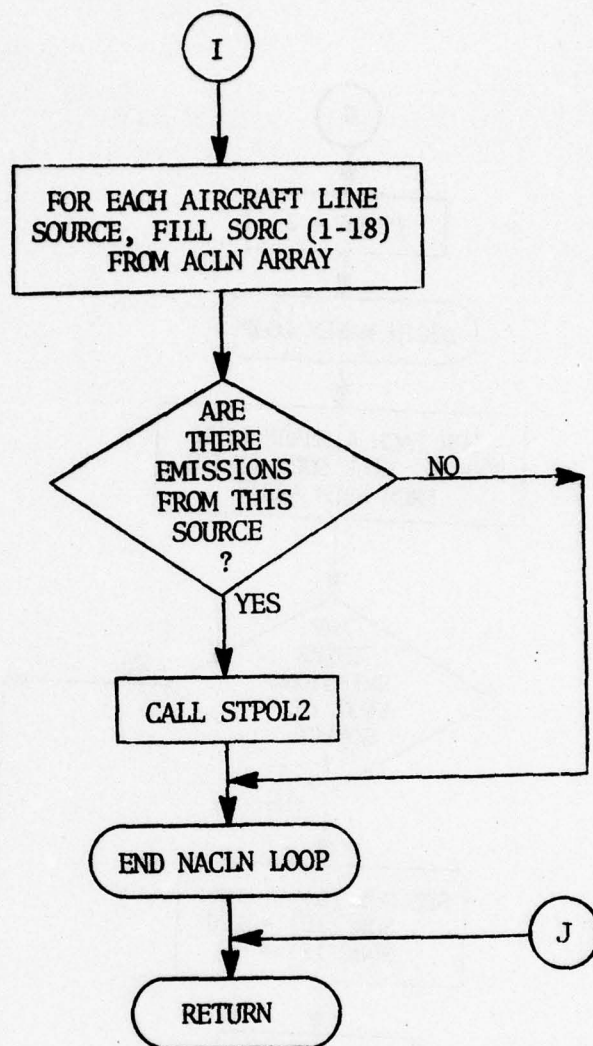












C	SUBROUTINE POLSOR	POLSR000
C	THIS ROUTINE IS THE DRIVER FOR THE DIFFUSION MODEL. FOR ALL	POLSR001
C	ENVIRON, AIRBASE AND AIRCRAFT POINTS, AREAS AND LINES,	POLSR002
C	THE SORC VECTOR IS FILLED WITH THE APPROPRIATE SOURCE	POLSR003
C	PARAMETERS AND THEN THE PROPER DIFFUSION ROUTINE IS CALLED	POLSR004
C		POLSR005
	COMMON /MET/ WS, WSMPI, IWS, WD, IWD, SINEWD, COSEWD, JSTAB, HLID, TEMP,	POLSR006
	. TEMK	POLSR007
	COMMON /SRCE/ NPOL, NENPT, NENAR, NENLN, NABPT, NABAR, NABLN, NACPT,	POLSR008
	. NACAR, NACLN, ENPT(16,100), ENAP(11,100), ENLN(14,20), ABPT(16,150),	POLSR009
	. ABAR(11,100), ABLN(14,100), ACPT(16,1), ACAR(11,24), ACLN(18,250)	POLSR010
	COMMON /INFO/ IRECEP, IWDIR, ITYPE, HTAERO, SORC(18), IPOL	POLSR011
	COMMON /AIRQAL/ RECDAT(3, 6, 312)	POLSR012
	COMMON /XTRAN/ XZ	POLSR013
C		POLSR014
C	SET RECDAT ARRAY TO 0.0	POLSR015
C		POLSR016
	DO 10 I=1,3	POLSR017
	DC 10 J=1,6	POLSR018
	DC 10 K=1,312	POLSR019
10	RECDAT(I,J,K) = 0.	POLSR020
	IFOL = NPOL	POLSR021
	HTAERO=20.*.3048	POLSR022
C		POLSR023
C	ENVIRON POINTS	POLSR024
C		POLSR025
	IF (NENPT .EQ. 0) GO TO 126	POLSR026
	ITYPE = 1	POLSR027
	DO 125 I=1, NENPT	POLSR028
	DC 124 J=1, 16	POLSR029
124	SCRC(J) = ENPT(J,I)	POLSR030
	DO 224 J=11, 16	POLSR031
	IF (SCRC(J) .NE. 0.0) GO TO 225	POLSR032
224	CONTINUE	POLSR033
	GO TO 125	POLSR034
225	CALL STPOL1	POLSR035
125	CONTINUE	POLSR036
C		POLSR037
C	AIRBASE POINTS	POLSR038
C		POLSR039
	126 IF (NABPT .EQ. 0) GO TO 136	POLSR040
	ITYPE = 2	POLSR041
	DO 135 I=1, NABPT	POLSR042
	DC 134 J=1, 16	POLSR043
134	SORC(J) = ABPT(J,I)	POLSR044
	DO 234 J=11, 16	POLSR045
	IF (SORC(J) .NE. 0.0) GO TO 235	POLSR046
234	CONTINUE	POLSR047
	GO TO 135	POLSR048
235	CALL STPOL1	POLSR049
135	CONTINUE	POLSR050
C		POLSR051
C	AIRCRAFT POINTS	POLSR052
C		POLSR053
	136 IF (NACPT .EQ. 0) GO TO 146	POLSR054
	ITYPE = 3	POLSR055
	DO 145 I=1, NACPT	POLSR056
	DC 144 J=1, 16	POLSR057
144	SCRC(J) = ACPT(J,I)	POLSR058
	DO 244 J=11, 16	POLSR059
	IF (SORC(J) .NE. 0.0) GO TO 245	POLSR060
		POLSR061

244	CONTINUE	POLSR062
	GO TO 145	POLSR063
245	CALL STPOL1	POLSR064
145	CCONTINUE	POLSR065
146	DC 150 I=1,18	POLSR066
150	SORC(I) = 0.	POLSR067
	SORC(10) = -1.	POLSR068
C		POLSR069
C	ENVIRON AREAS	POLSR070
C		POLSR071
	IF (NENAR .EQ. 0) GO TO 156	POLSR072
	ITYPE = 1	POLSR073
	DO 155 I=1,NENAR	POLSR074
	DO 153 J=1,5	POLSR075
153	SORC(J) = ENAR(J,I)	POLSR076
	DO 154 J=6,11	POLSR077
154	SORC(J+5) = ENAR(J,I)	POLSR078
	DO 253 J=11,16	POLSR079
	IF(SORC(J).NE.0.0) GO TO 254	POLSR080
253	CCONTINUE	POLSR081
	GO TO 155	POLSR082
254	CALL STPOL1	POLSR083
155	CONTINUE	POLSR084
C		POLSR085
C	AIREASE AREAS	POLSR086
C		POLSR087
	156 IF (NABAR .EQ. 0) GO TO 166	POLSR088
	ITYPE = 2	POLSR089
	DO 165 I=1,NABAR	POLSR090
	DC 163 J=1,5	POLSR091
163	SORC(J) = ABAR(J,I)	POLSR092
	DO 164 J=6,11	POLSR093
164	SORC(J+5) = ABAR(J,I)	POLSR094
	DO 263 J=11,16	POLSR095
	IF(SORC(J).NE.0.0) GO TO 264	POLSR096
263	CONTINUE	POLSR097
	GO TO 165	POLSR098
264	CALL STPOL1	POLSR099
165	CCONTINUE	POLSR100
C		POLSR101
C	AIRCRAFT AREAS	POLSR102
C		POLSR103
	166 IF (NACAR .EQ. 0) GO TO 176	POLSR104
	ITYPE = 3	POLSR105
	DO 175 I=1,NACAR	POLSR106
	DO 173 J=1,5	POLSR107
173	SORC(J) = ACAR(J,I)	POLSR108
	DO 174 J=6,11	POLSR109
174	SORC(J+5) = ACAR(J,I)	POLSR110
	DO 273 J=11,16	POLSR111
	IF(SORC(J).NE.0.0) GO TO 274	POLSR112
273	CCONTINUE	POLSR113
	GO TO 175	POLSR114
274	CALL STPOL1	POLSR115
175	CCONTINUE	POLSR116
C		POLSR117
C	CRITICAL DISTANCE, XZ, MUST BE CONVERTED TO METERS FOR LINE MODEL	POLSR118
C		POLSR119
	176 XZ = XZ * 1000.	POLSR120
C		POLSR121
C	ENVIRON LINES	POLSR122
C		POLSR123



```

      IF (NENLN .EQ. 0) GO TO 186
      ITYPE = 1
      DC 185 I=1,NENLN
      DC 184 J=1,8
184  SOPC(J) = ENLN(J,I)
      DO 384 J=13,18
384  SCRC(J)=ENLN(J-4,I)
      DO 284 J=13,18
      IF(SCRC(J).NE.0.0) GO TO 285
284  CONTINUE
      GO TO 185
285  SCRC( 9) = 100.
      SCRC(10) = 100.
      SCRC(12) = 1.
      CALL STPOL2
185  CONTINUE
C
C  AIRBASE LINES
C
186  IF(NABIN .EQ. 0) GO TO 196
      ITYPE = 2
      DC 195 I=1,NABLN
      DC 194 J=1,8
194  SCRC(J) = ABLN(J,I)
      DO 394 J=13,18
394  SCRC(J)=ABLN(J-4,I)
      DO 294 J=13,18
      IF(SCRC(J).NE.0.0) GO TO 295
294  CCNTINUE
      GO TO 195
295  SCRC( 9) = 100.
      SCRC(10) = 100.
      SCRC(12) = 1.
      CALL STPOL2
195  CONTINUE
C
C  AIRCRAFT LINES
C
196  IF (NACLN .EQ. 0) GO TO 206
      ITYPE = 3
      DO 205 I=1,NACLN
      DO 204 J=1,18
204  SCRC(J) = ACLN(J,I)
      DO 304 J=13,18
      IF(SCRC(J).NE.0.0) GO TO 305
304  CCNTINUE
      GO TO 205
305  CALL STPOL2
205  CONTINUE
206  CCNTINUE
      RETURN
      END

```

```

POLSR124
POLSR125
POLSP126
POLSR127
POLSR128
POLSR129
POLSR130
POLSR131
POLSR132
POLSR133
POLSR134
POLSR135
POLSR136
POLSR137
POLSR138
POLSR139
POLSR140
POLSR141
POLSR142
POLSR143
POLSR144
POLSR145
POLSR146
POLSR147
POLSR148
POLSR149
POLSR150
POLSR151
POLSR152
POLSR153
POLSR154
POLSR155
POLSR156
POLSR157
POLSR158
POLSR159
POLSR160
POLSR161
POLSR162
POLSR163
POLSR164
POLSR165
POLSR166
POLSR167
POLSR168
POLSR169
POLSR170
POLSR171
POLSR172
POLSR173
POLSR174
POLSR175

```

## SUBROUTINE PSEUDO

### Purpose:

To call the SIGCY and SIGCZ functions to find the virtual distance in meters from the source to the pseudo upwind point source.

### Input:

1. Initial dispersions in y and z directions.
2. Wind speed and stability class.

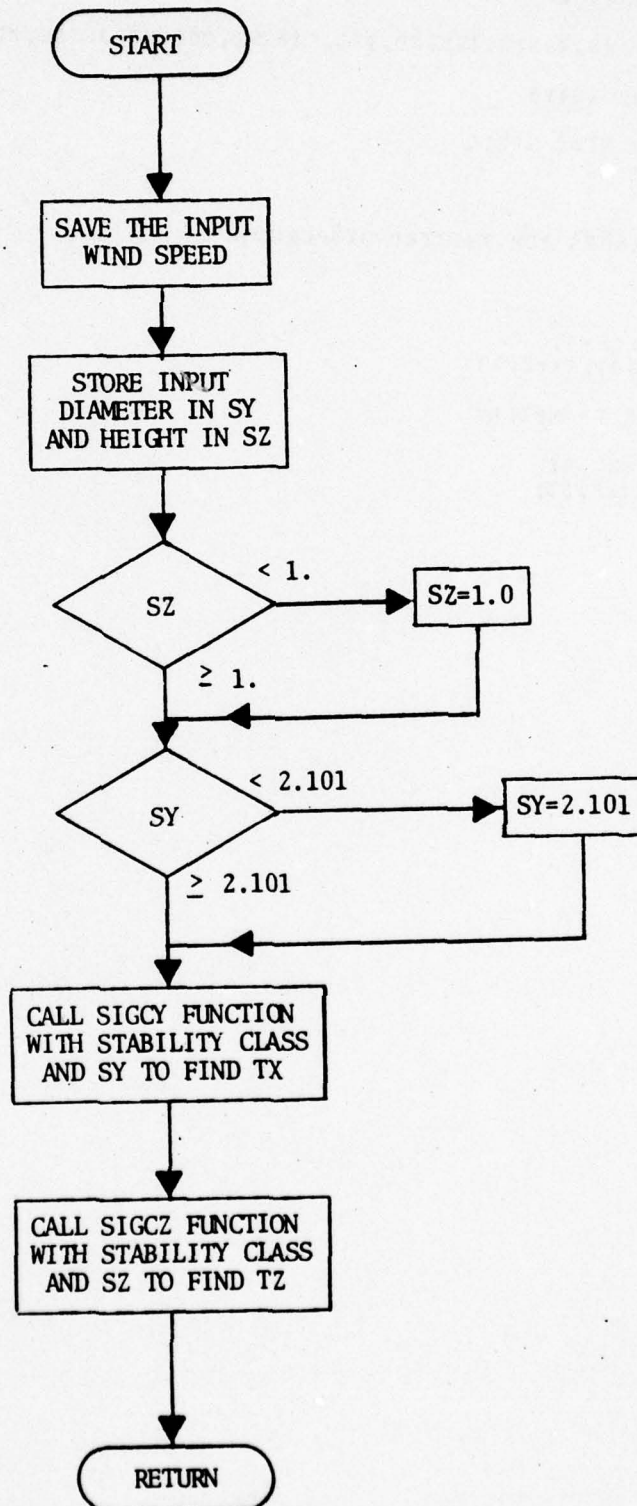
### Output:

The virtual distances in the y and z directions.

### Functions Called:

SIGCY  
SIGCZ

SUBROUTINE PSEUDO





	SUBROUTINE PSEUDO(DS,WIN,HS,TY,TZ)	PSUD0000
C		PSUD0001
C	THIS SUBROUTINE CALLS THE SIGCY AND SIGCZ FUNCTIONS	PSUD0002
C	TO FIND THE VIRTUAL DISTANCE FROM THE SOURCE TO THE PSEUDO	PSUD0003
C	UPWIND POINT SOURCE	PSUD0004
C		PSUD0005
	COMMON /MET/ WS,WSMPH,IWS,WD,TWD,SINEWD,COSEWD,JSTAB,HLID,	PSUD0006
	. TEMP,TEHK	PSUD0007
	COMMON /WDUN/ WSAVE	PSUD0008
C		PSUD0009
C	SAVE THE INPUT WIND SPEED	PSUD0010
C		PSUD0011
	WSAVE=WIN	PSUD0012
C		PSUD0013
C	SET MINIMUM VALUES FOR INITIAL DISPERSIONS	PSUD0014
C		PSUD0015
	SY=DS	PSUD0016
	SZ=HS	PSUD0017
	IF(SZ.LT.1.) SZ=1.0	PSUD0018
	IF(SY.LT.2.101) SY=2.101	PSUD0019
C		PSUD0020
C	FIND DISTANCES IN METERS	PSUD0021
C		PSUD0022
	TY=SIGCY(JSTAB,SY)	PSUD0023
	TZ=SIGCZ(JSTAB,SZ)	PSUD0024
	RETURN	PSUD0025
	END	PSUD0026

## SUBROUTINE QMOD

### Purpose:

To compute the linear distribution, in inverse length, of the pollution along a runway due to aircraft emission during landing or takeoff.

### Input:

YSI	Distance along runway measured from tip of exhaust plume near starting end of runway
TAIL	Length or penetration of exhaust plume of aircraft at rest
DL	Length of smoke slug on runway
A	Acceleration (or deceleration) of aircraft
VI2	Initial velocity squared
VS	Average velocity of exhaust particles relative to air mass in exhaust plume
WS2	Wind speed squared
WSC	2·wind speed·(- cosine of angle between runway and wind vector)
RR	A/G, where A is acceleration and G is the normalization constant for line density

### Output:

QL	The linear distribution of pollution
----	--------------------------------------

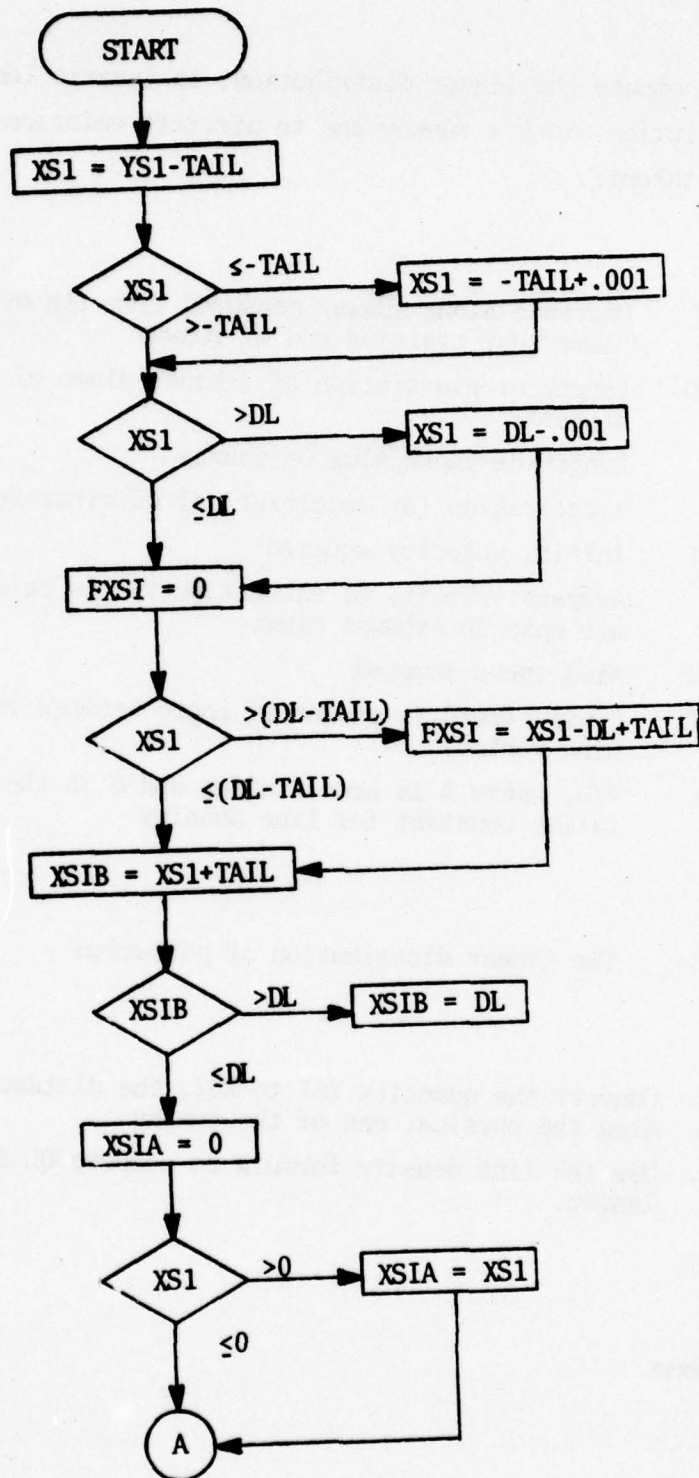
### Procedure:

1. Convert the quantity YSI to XSI, the distance measured from the physical end of the runway.
2. Use the line density formula to compute QL in inverse length.

### Subroutines Called:

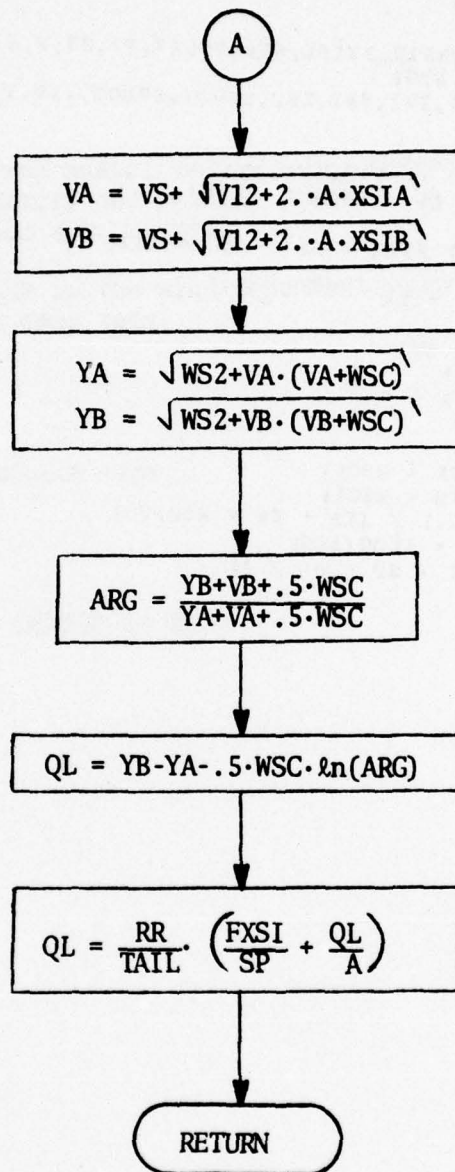
None

SUBROUTINE QMOD





SUBROUTINE QMOD (Contd.)



	SUBROUTINE QMOD (YS1,QL)	QMOD0000
C		QMOD0001
C	THIS ROUTINE COMPUTES THE LINEAR DISTRIBUTION, IN INVERSE LENGTH,	QMOD0002
C	OF THE POLLUTION ALONG A RUNWAY DUE TO AIRCRAFT EMISSION	QMOD0003
C	DURING LANDING OR TAKEOFF	QMOD0004
C		QMOD0005
	COMMON /INFO/ IRECEP,IWDIR,ITYPE,HTAERO,X1,Y1,Z1,W,DELZ,X2,Y2,Z2,	QMOD0006
	. V1,V2,DL,TIME,EMIS(6),NPOL	QMOD0007
	COMMON /LN/ XW1,YW1,ZW1,XW2,YW2,ZW2,SUDOY,SUDOZ,IAD,TAIL,A,V12,VS,	QMOD0008
	. WS2,WSC,RR,SP	QMOD0009
	XS1 = YS1 - TAIL	QMOD0010
	IF (XS1 .LE. -TAIL) XS1 = -TAIL + .001	QMOD0011
	IF (XS1 .GT. DL) XS1 = DL - .001	QMOD0012
	PXSI = 0.	QMOD0013
	IF (XS1 .GT. (DL-TAIL)) PXSI = XS1 - DL + TAIL	QMOD0014
30	XSIB = XS1 + TAIL	QMOD0015
	IF (XSIB .GT. DL) XSIB = DL	QMOD0016
	XSIA = 0.	QMOD0017
	IF (XS1 .GT. 0) XSIA = XS1	QMOD0018
	ROOTB = V12 + 2.*A*XSIB	QMOD0019
	ROOTA = V12 + 2.*A*XSIA	QMOD0020
	VA = SQRT(ROOTA) + VS	QMOD0021
	VB = SQRT(ROOTB) + VS	QMOD0022
	YA = SQRT(WS2 + VA *(VA + WSC))	QMOD0023
	YB = SQRT(WS2 + VB *(VB + WSC))	QMOD0024
	ARG = (YB + VB + WSC/2.) / (YA + VA + WSC/2.)	QMOD0025
	QL = YB - YA - WSC/2. * ALOG(ARG)	QMOD0026
	QL = RR / TAIL * (PXSI / SP + QL / A)	QMOD0027
	RETURN	QMOD0028
	END	QMOD0029

## SUBROUTINE READ

### Purpose:

1. To read master source tape, thereby providing the emission inventory and related data to the source emission distribution subroutines.
2. To set up the wind-dependent sources as random access disk data sets.

### Input:

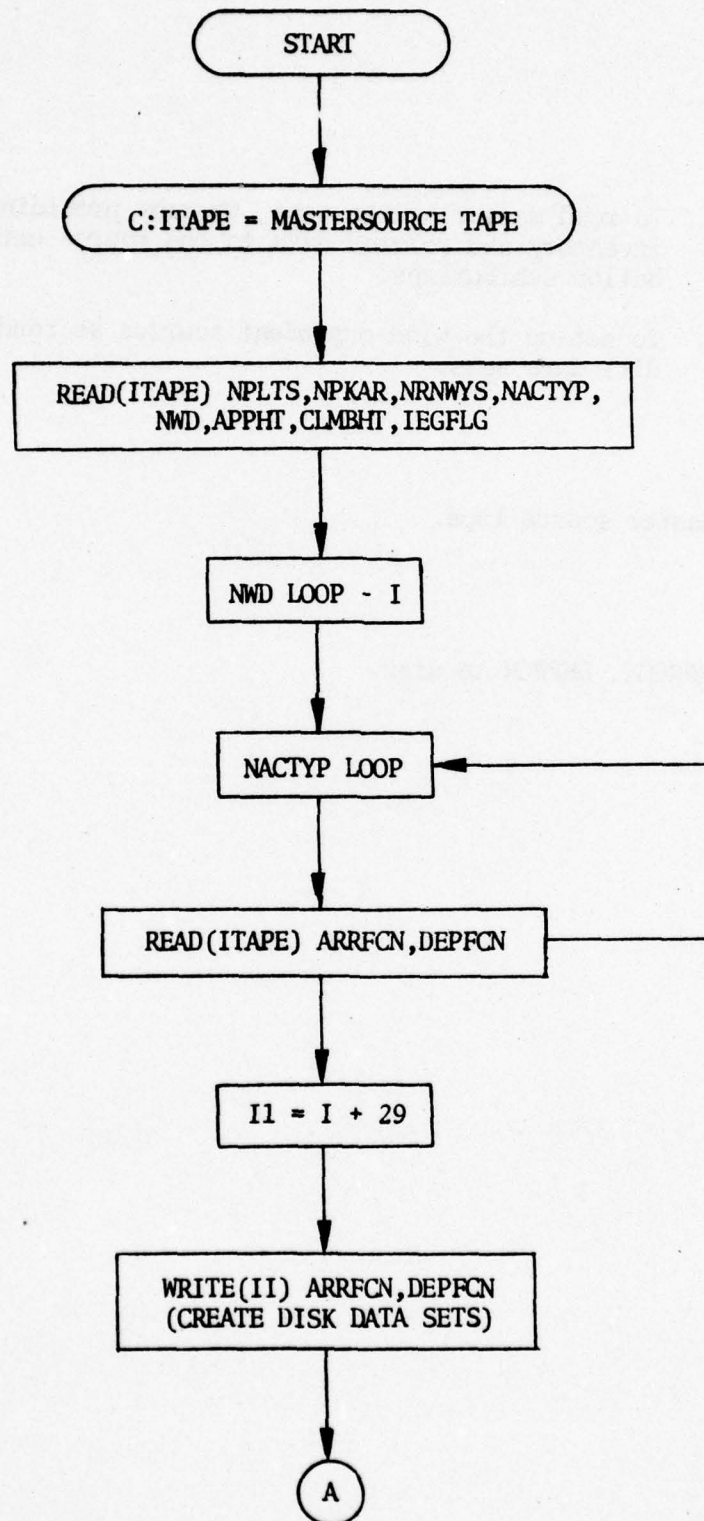
Master source tape.

### Output:

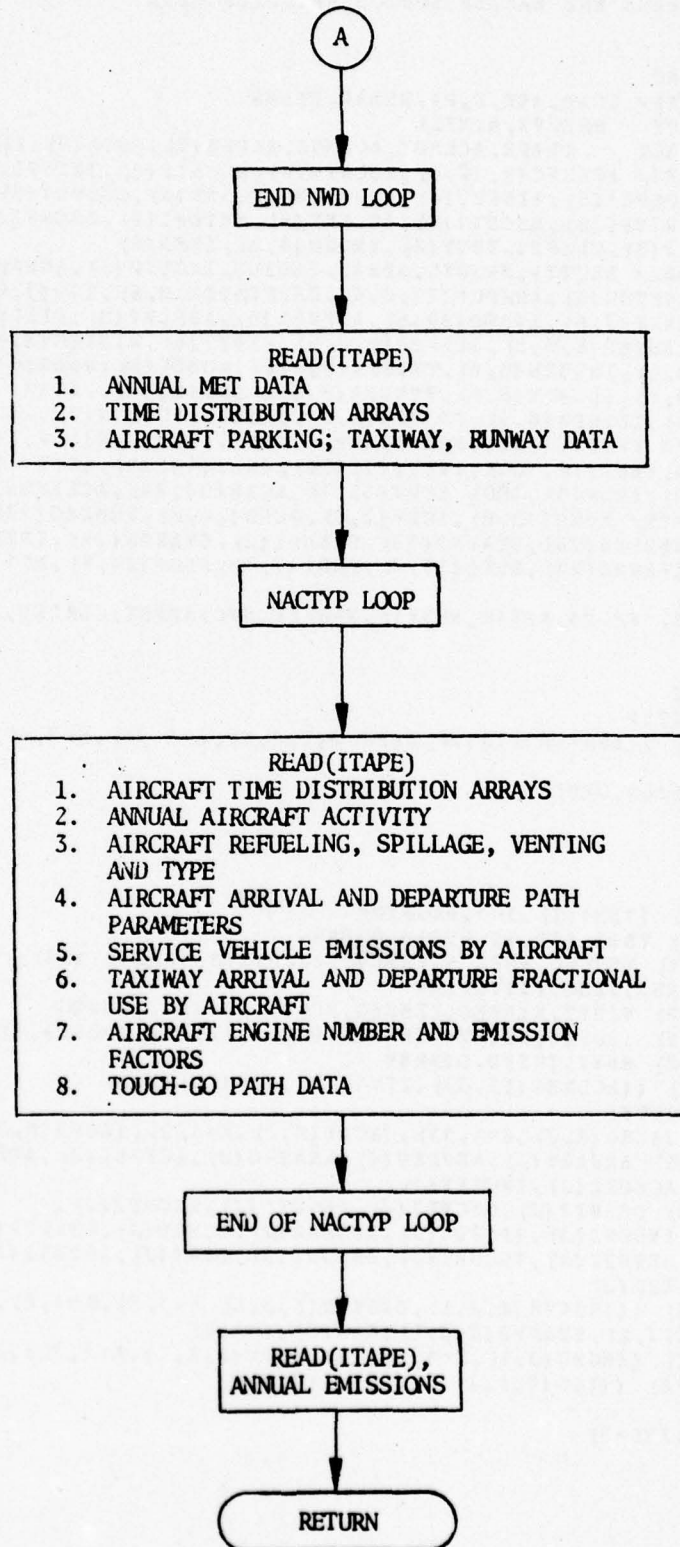
ARRFCN, DEPFEN to disk.



SUBROUTINE READ



SUBROUTINE READ (Continued)



	SUBROUTINE READ	PEAD0000
C		READ0001
C	THIS ROUTINE READS THE MASTER SOURCE EMISSION TAPE	READ0002
C		READ0003
	REAL LNDSPD	READ0004
	INTEGER ENGNO	READ0005
	COMMON /ANNMET/ TBAR,ACD,P,PA,WSBAR,DTBAR	READ0006
	COMMON / RECP / MRECPT,MAXFIL	READ0007
	COMMON / DEFAULT / ITAPE,ACLNDY,ACLNDZ,ALPHA(7),BETA(7),FLDENS(7)	READ0008
	COMMON /ACEDB1/ ACEMFC(8,10,6),ASCNT1(8),ASCNT2(8),TXISPD(8),	READ0009
	. LNDSFD(8),APSPD1(8),AFSPD2(8),COHT1(8),TOSPD(8),COSPD1(8),	READ0010
	. COSFD2(8),SRTUPT(8),DSCNT1(8),EGCHKT(8),SHTDNT(8),DSCNT2(8),	READ0011
	. APPHT,APPHT2(8),CLMBHT,TOWT(8),ENGNO(8,2),IDRR(8)	READ0012
	COMMON /ACEDB2/ NACTYP,NRNWYS,NPKAR,IEGFLG,IACTYP(8),ANNAAR(8),	READ0013
	. ANNLEP(8),ANNTGO(8),ARRFCN(24,8,6),DEPPCN(24,8,6),TGO(3,4,8),	READ0014
	. DISRNW(6),RNWY(7,6),IUSWD(20,6),ACFUEL(8),ARFLVT(8),DPFLVT(8),	READ0015
	. ACSFIL(8),ARSVEM(6,8,5),DPSVEM(6,8,5),NIBTT(6),NIBSEG(8,6),	READ0016
	. IIBSEG(16,8,6),IDIBTW(8,6),TTARFR(8,8,6),NOBTT(6),NOBSEG(8,6),	READ0017
	. IOBSEG(16,8,6),IDOBTW(8,6),TDPFR(8,8,6),NPASQ(6),IDPRKA(6),	READ0018
	. PAREA(6,3,3),IDIBPA(8,6),IDOBPA(8,6),NLSEGS,ACLNSG(12,25),JES1(8)	READ0019
	COMMON /SRCE/ NPLTS,NENPT,NENAR,NENLN,NABPT,NABAR,NABLN,NACPT,	READ0020
	. NACAR,NACLN,ENET(16,100),ENAR(11,100),ENLN(14,20),ABPT(16,150),	READ0021
	. ABAR(11,100),ABLN(14,100),ACPT(16,1),ACAR(11,24),ACLN(18,250)	READ0022
	COMMON /DSTRET/ ACHO(13,8),ACDY(2,8),ACHR(24,8),VHMLMO(13),	READ0023
	. VHMLDY(2),VHMLHR(24),CVABMO(13),CVABDY(2),CVABHR(24),CVENMO(13),	READ0024
	. CVENDY(2),CVENHR(24),FLMO(13,7),FLDY(2,7),FLHR(24,7),NC1	READ0025
C		READ0026
	REAL (ITAPE) NPLTS,NPKAR,NRNWYS,NACTYP,NWD,APPHT,CLMBHT,IEGFLG	READ0027
	. , NISEGS	READ0028
	REWIND 30	READ0029
	DC 2 I=1,NWD	READ0030
	DC 5 J=1,NACTYP	READ0031
	READ (ITAPE) ((ARRFCN(L,J,K),DEPPCN(L,J,K),L=1,24),K=1,6)	READ0032
5	CCONTINUE	READ0033
	WRITE(30)ARRFCN,DEPPCN	READ0034
2	CCONTINUE	READ0035
	REWIND 30	READ0036
	MRECPT=1	READ0037
	MAXFIL=NWD	READ0038
	READ (ITAPE) (JES1(I),I=1,NACTYP)	READ0039
	READ (ITAPE) TBAR,ADD,PA,WSBAR,DTBAR	READ0040
	READ (ITAPE) VHMLMO,VHML ≤,VHMLqr≤ceabmO≤ceaaaH,c aaqr≤c enM LT	READ0041
	. CVENDY,CVENHR,FLMO,FLDY,FLHR	READ0042
	READ (ITAPE) NIBTT,NIBSEG,IIBSEG,NOBTT,NOBSEG,IOBSEG	READ0043
	READ (ITAPE) IDOBTW,ICIBTW,IDPRKA,PAREA,IDIBPA,IDOBPA,NPASQ	READ0044
	READ (ITAPE) RNWY,IUSWD,DISRNW	READ0045
	READ (ITAPE) ((ACLNSG(II,JJ),II=1,12),JJ=1,NLSEGS)	READ0046
	DC 40 J=1,NACTYP	READ0047
	READ (ITAPE) (ACHO(K,J),K=1,13), (ACDY(K,J),K=1,2), (ACHR(K,J),K=1,24)	READ0048
	READ (ITAPE) ANNARR(J),ANNDEP(J),ANNTGO(J),ACFUEL(J),ARFLVT(J),	READ0049
	. DPFLVT(J),ACSPIL(J),IACTYP(J)	READ0050
	READ (ITAPE) DSCNT1(J),DSCNT2(J),ASCNT1(J),ASCNT2(J),	READ0051
	. TXISED(J),LNDSFD(J),APSPD1(J),APSPD2(J),TOSPD(J),COSPD1(J),	READ0052
	. COSFD2(J),SRTUPT(J),EGCHKT(J),SHTDNT(J),TOWT(J),APPHT2(J),	READ0053
	. COHT1(J),IDRR(J)	READ0054
	READ (ITAPE) ((ARSVEM(K,J,L),DPSVEM(K,J,L),L=1,5),K=1,6),	READ0055
	. ((TTARFR(K,J,L),TDPFR(K,J,L),K=1,8),L=1,6)	READ0056
	READ (ITAPE) (ENGNO(J,L),L=1,2), ((ACEMFC(J,K,L),K=1,10),L=1,6)	READ0057
	READ (ITAPE) ((TGO(K,L,J),K=1,3),L=1,4)	READ0058
40	CCONTINUE	READ0059
	4 READ (ITAPE,END=3)	READ0060
	GO TO 4	READ0061



3 CCNTINUE  
RETURN  
END

READ0062  
READ0063  
READ0064

## FUNCTION RISE

Purpose:

To calculate the plume rise using either the Carson-Moses or Holland plume rise formula.

Input:

Stack parameters, current wind speed and stability, temperature, and the plume rise flag.

Output:

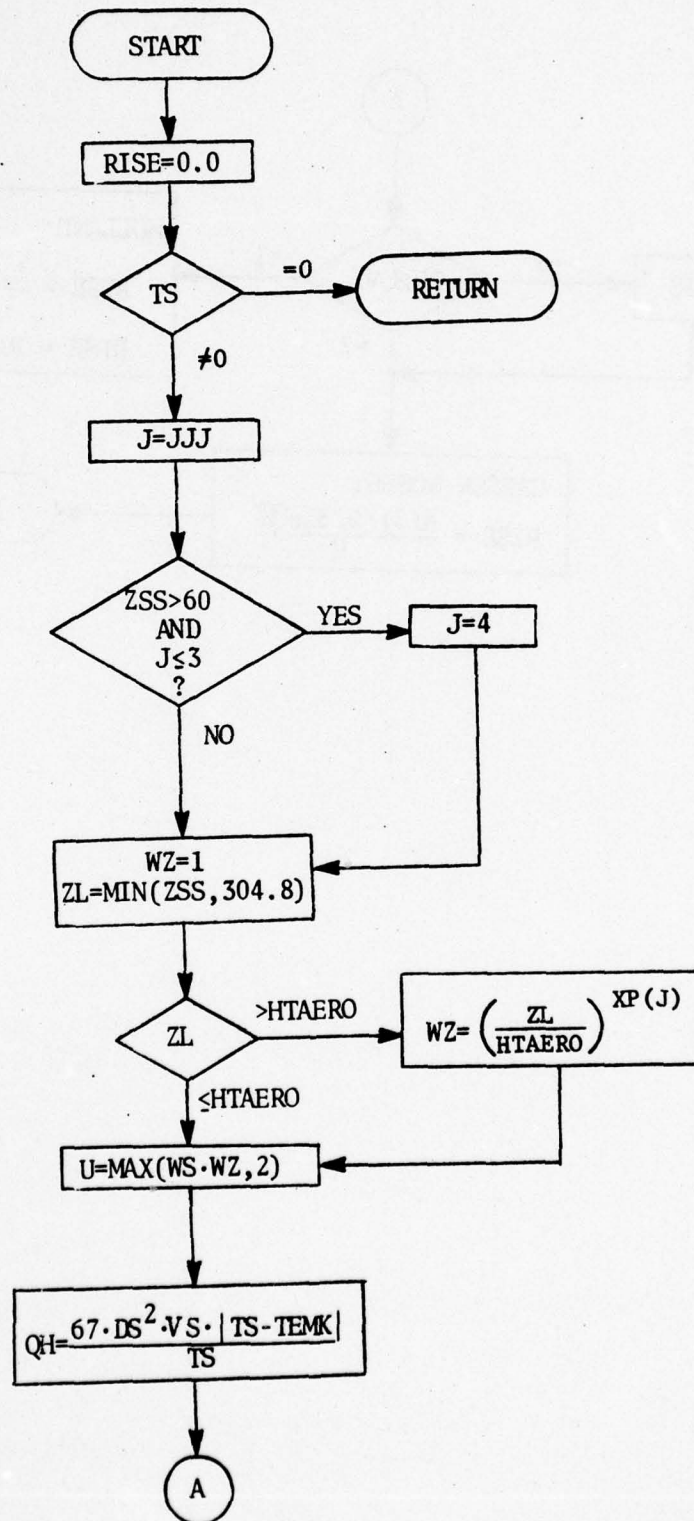
The height of the plume rise.

Subroutines

Called:

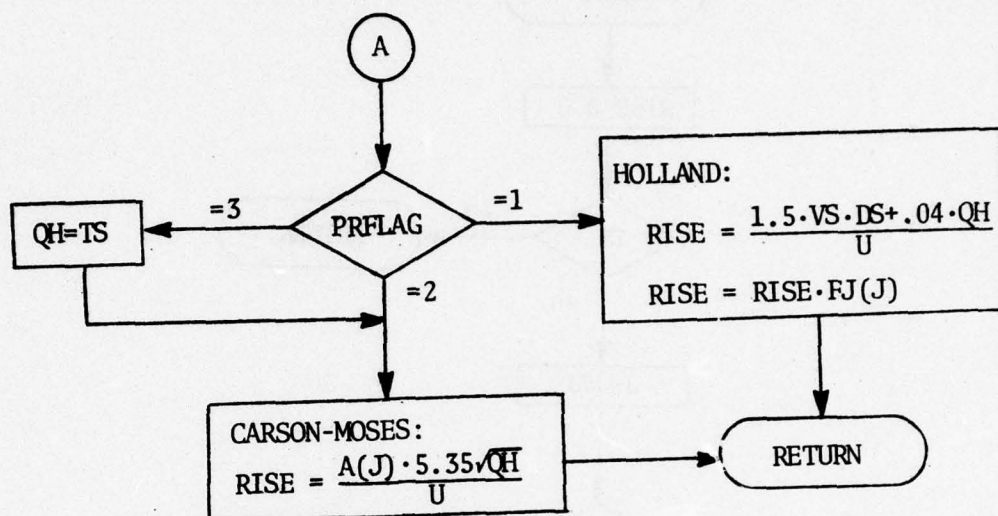
None

# FUNCTION RISE





FUNCTION RISE (Cont'd.)



C	FUNCTION RISE(ZSS,JJJ)	RISE0000
C	THIS FUNCTION CALCULATES THE PLUME RISE	RISE0001
C	ZSS IS THE PHYSICAL STACK HEIGHT MODIFIED FOR DOWNWASH	RISE0002
C	EFFECTS, IF ANY	RISE0003
C	JJJ IS THE AMBIENT STABILITY	RISE0004
C	COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,	RISE0005
	TEMK	RISE0006
	COMMON /INFO/ IRECEP,IWDIR,ITYPE,HTAERO,XS,YS,ZS,DELY,DELZ,	RISE0007
	TS,VS,DS,HB,PRFLAG,EMIS(8),NPOL	RISE0008
	DIMENSION A(6),FJ(6)	RISE0009
	COMMON /WINDPRO/ XP(6)	RISE0010
	DATA A /2.65,2.65,2.65,1.08,2*0.68/,	RISE0011
	FJ / 1.2,1.2,1.2,1.0,0.8,0.8/	RISE0012
	RISE=0.0	RISE0013
C	CHECK THE STACK EXIT GAS TEMPERATURE	RISE0014
C	IF (TS.EQ.0.0) RETURN	RISE0015
C	FOR TALL STACKS USE STABILITY 4 IN THE WIND PROFILE LAW	RISE0016
C	J=JJJ	RISE0017
C	IF (ZSS.GT.60.AND.J.LE.3) J=4	RISE0018
C	COMPUTE THE WIND SPEED AT THE ELEVATION OF THE STACK	RISE0019
C	FOR STABILITY J	RISE0020
C	WZ=1.0	RISE0021
	ZL=AMIN1(ZSS,304.8)	RISE0022
	IF (ZL.GT.HTAERO) WZ= (ZL/HTAERO)**XP(J)	RISE0023
	U=AMAX1(WS*WZ,2.0)	RISE0024
C	COMPUTE THE THERMAL EMISSION RATE	RISE0025
C	QH=67.0*DS*DS*VS*ABS(TS-TEMK)/TS	RISE0026
C	IF (PRFLAG.EQ.1.0) GO TO 1	RISE0027
C	IF (PRFLAG.EQ.3.0) QH=TS	RISE0028
C	CARSON-MOSES PLUME RISE FORMULA	RISE0029
C	RISE=A(J)*5.35*SQRT(QH)/U	RISE0030
C	RETURN	RISE0031
C	HOLLAND PLUME RISE FORMULA	RISE0032
C	1 CONTINUE	RISE0033
	RISE=1.5*VS*DS/U+0.04*QH/U	RISE0034
	RISE=RISE*FJ(J)	RISE0035
	RETURN	RISE0036
	END	RISE0037
		RISE0038
		RISE0039
		RISE0040
		RISE0041
		RISE0042
		RISE0043
		RISE0044
		RISE0045
		RISE0046
		RISE0047
		RISE0048
		RISE0049
		RISE0050
		RISE0051

## FUNCTION RRDIST

### Purpose:

To calculate the length of runway necessary for takeoff using aircraft dependent equations.

### Input:

Aircraft identification, pressure altitude, ambient temperature and wind speed, and aircraft takeoff weight.

### Output:

Takeoff length in feet of runway roll to liftoff.

### Procedure:

For a given aircraft, use the proper set of takeoff equations provided by the USAF.

### Subroutines Called:

None



C	FUNCTION ERDIST (IR,PA,T,GW,WS)	RRDST000
C	FUNCTION CALCULATES RUNWAY ROLL DISTANCE IN FEET	RRDST001
C	IR IS AIRCRAFT IDENTIFICATION NUMBER	RRDST002
C	PA IS PRESSURE ALTITUDE IN HUNDREDS OF FEET	RRDST003
C	T IS TEMPERATURE IN DEGREES FAHRENHEIT	RRDST004
C	GW IS AC TAKE OFF WEIGHT IN THOUSAND POUNDS	RRDST005
C	WS IS THE WIND SPEED IN KNOTS	RRDST006
C		RRDST007
C		RRDST008
	FGR=0.0	RRDST009
	IF (IR.EQ.100) GO TO 100	RRDST010
	GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,	RRDST011
	123,24,25,26,27,28,29,30,31,32,33,34,35,36,37,100,100,100,100,100,	RRDST012
	2 100,100,100,100,100,100,100,12),IR	RRDST013
1	CCNTINUE	RRDST014
	GO TO 100	RRDST015
3	CCNTINUE	RRDST016
2	TCF=-(2.78-8.5714E-4*PA)+(1.82E-2+7.2857E-5*PA)*GW	RRDST017
	GR=(1.184E+1-4.2167E-1*T+1.0E-2*T**2-4.583E-5*T**3)+	RRDST018
	(4.194+1.7197E-2*T-9.26018E-4*T**2)*TOF+	RRDST019
	(1.0457+8.40E-3*T+2.117E-4*T**2+2.98E-7*T**3)*TOF**2	RRDST020
	FGR=(GR-(1.15E-1+9.0E-3*GR)*WS)*100.	RRDST021
	GO TO 100	RRDST022
4	CCNTINUE	RRDST023
5	TOF=(1.589+6.883E-3*PA+1.2767E-4*PA**2)+	RRDST024
	(6.819E-3+1.1007E-4*PA-3.924E-7*PA**2)*T+	RRDST025
	(5.979E-5+3.38096E-7*PA+8.532E-9*PA**2)*T**2	RRDST026
	GR=(-13.25+8.75E-1*GW-1.25E-2*GW**2)+	RRDST027
	(1.3925E+1-9.275E-1*GW+2.125E-2*GW**2)*TOF	RRDST028
	FGR=(GR-(1.316E-1+8.748E-3*GR)*WS)*100.	RRDST029
	GO TO 100	RRDST030
6	TOF=(9.3937E-1+2.0947E-2*PA+2.005E-4*PA**2)+	RRDST031
	(3.746467E-2+4.05625E-4*PA)*T+	RRDST032
	(1.9928E-4-5.75006E-6*PA+1.40234E-7*PA**2)*T**2	RRDST033
	GR=(1.4307E+1-7.57144E-1*GW+2.6785E-2*GW**2)+	RRDST034
	(1.67257E+1-1.17762*GW+2.7381E-2*GW**2)*TOF	RRDST035
	FGR=(GR-(2.412799E-2+7.82971E-3*GR)*WS)*100.	RRDST036
	GO TO 100	RRDST037
7	TOF=(-1.06E-3+1.674E-2*PA+8.1888E-5*PA**2)+	RRDST038
	(1.36E-2+9.592E-6*PA+1.755E-6*PA**2)*T+	RRDST039
	(5.1099E-5+1.2899E-6*PA-6.123E-9*PA**2)*T**2	RRDST040
	GR=(-1.423E+1+6.349998E-1*GW+1.6667E-3*GW**2)+	RRDST041
	(6.1857-3.2179E-1*GW+8.214E-3*GW**2)*TOF	RRDST042
	FGR=(GR-(6.293E-2+7.328E-3*GR)*WS)*100.	RRDST043
	GO TO 100	RRDST044
8	TOF=(9.503E-2+3.313E-2*PA+1.3666E-4*PA**2)+	RRDST045
	(2.2546E-2+1.7848E-4*PA-4.04E-6*PA**2)*T+	RRDST046
	(1.3438E-4-1.2166E-6*PA+4.1854E-8*PA**2)*T**2	RRDST047
	GR=(2.95E+1-2.394*GW+6.497E-2*GW**2)+	RRDST048
	(3.1035+7.52E-2*GW-3.186E-3*GW**2)*TOF+	RRDST049
	(1.2715-1.5535E-1*GW+4.3889E-3*GW**2)*TOF**2	RRDST050
	FGR=(GR-(9.0E-2+1.807E-2*GR-7.143E-5*GR**2)*WS)*100.	RRDST051
	GO TO 100	RRDST052
9	TCF=(3.36455E-3+5.63556E-2*PA)+	RRDST053
	(4.417E-2-2.031E-3*PA+5.63E-5*PA**2-3.9954E-7*PA**3)*T+	RRDST054
	(-9.2E-5+2.08E-5*PA-5.39E-7*PA**2+3.8E-9*PA**3)*T**2	RRDST055
	GR=(1.65838-3.069E-1*GW+8.1363E-2*GW**2)+	RRDST056
	(-3.6111+3.63559E-1*GW)*TOF+	RRDST057
	(7.3975E-1-8.78749E-2*GW+3.2487E-3*GW**2)*TOF**2	RRDST058
	FGR=(GR-(5.0E-2+7.4E-3*GR)*WS)*100.	RRDST059
	GO TO 100	RRDST060
10	TOF=(12.5546-5.7192E-2*PA+1.3075E-4*PA**2)-	RRDST061

•	(2.9032E-2-1.0254E-4*PA-1.45125E-7*PA**2)*T	RRDST062
•	GR=(-5.14955E+1+2.57957*GW-1.4425E-2*GW**2)-	RRDST063
•	(-1.1535E+1+5.915E-1*GW-4.6828E-3*GW**2)*TOF+	RRDST064
•	(-6.2285E-1+3.2375E-2*GW-2.9056E-4*GW**2)*TOF**2)*1000.	RRDST065
•	FGR=(3.305E+1+9.729E-1*GR+2.31E-6*GR**2)-	RRDST066
•	(8.244+8.3598E-3*GR-1.44E-8*GR**2)*WS	RRDST067
	GO TO 100	RRDST068
11	TCF=(7.436E-1+4.29E-2*PA)+(2.1276E-2-3.1116E-5*PA)*T	RRDST069
	GR=(1.638E+1-7.78E-1*GW+2.84E-2*GW**2)+	RRDST070
•	(3.809-1.947E-1*GW+4.264E-3*GW**2)*TOF+	RRDST071
•	(-1.976E-1+1.5757E-2*GW+4.6189E-4*GW**2)*TOF**2	RRDST072
•	FGR=(GR-(8.5E-2+8.25E-3*GR)*WS)*100.	RRDST073
	GO TO 100	RRDST074
12	TCF=(1.1405-4.659E-3*PA+1.28E-5*PA**2)-	RRDST075
•	(2.0146E-3-2.46E-5*PA+3.5514E-7*PA**2)*T	RRDST076
•	GR=(-3.0029E+1-9.6225E-2*GW+1.25428E-1*GW**2)-	RRDST077
•	(-7.3845E+1+1.20433*GW+1.7857E-1*GW**2)*TOF+	RRDST078
•	(-3.57857E+1+7.857E-1*GW+7.14286E-2*GW**2)*TOF**2	RRDST079
•	FGR=((3.17413E-1+9.762E-1*GR+2.657E-4*GR**2)-	RRDST080
•	(1.1114E-1+7.91177E-3*GR+4.40169E-5*GR**2)*WS)*100.	RRDST081
	GO TO 100	RRDST082
13	TOF=(9.166-5.485E-2*PA)-(3.412E-2-1.8E-4*PA)*T	RRDST083
•	GR=(3.02E+2-3.519E+1*GW+1.841*GW**2)-	RRDST084
•	(1.306E+2-1.277E+1*GW+5.4E-1*GW**2)*TOF+	RRDST085
•	(2.0687E+1-1.715*GW+6.07E-2*GW**2)*TOF**2-	RRDST086
•	(1.1578-8.4228E-2*GW+2.46E-3*GW**2)*TOF**3	RRDST087
•	FGR=(GR-(9.55E-2+7.15E-3*GR)*WS)*100.	RRDST088
	GO TO 100	RRDST089
14	TOF=(2.336+1.582E-2*PA+1.172E-4*PA**2)+	RRDST090
•	(5.604E-3+9.97746E-5*PA-5.8117147E-7*PA**2)*T+	RRDST091
•	(9.19269E-5-1.34357E-8*PA+1.61411E-8*PA**2)*T**2	RRDST092
•	GR=(7.7366-2.52997E-1*GW+2.385E-3*GW**2)+	RRDST093
•	(-2.1071+4.2586E-2*GW+12.748E-4*GW**2)*TOF	RRDST094
•	FGR=(GR-(1.0755E-1+1.4588E-2*GR-7.94156E-5*GR**2)*WS)*100.	RRDST095
	GO TO 100	RRDST096
15	CONTINUE	RRDST097
	GO TO 100	RRDST098
16	TOF=(7.6859-1.15E-1*PA+4.413E-4*PA**2)-	RRDST099
•	(2.925E-2-8.1128E-4*PA+6.999E-6*PA**2)*T-	RRDST100
•	(2.2289E-4+5.054E-6*PA-7.57E-8*PA**2)*T**2	RRDST101
•	GR=(2.546E+1-2.3388*GW+1.0717E-1*GW**2)-	RRDST102
•	(7.9095-6.7434E-1*GW+2.1045E-2*GW**2)*TOF+	RRDST103
•	(6.099E-1-5.0858E-2*GW+1.434E-3*GW**2)*TOF**2	RRDST104
•	FGR=(GR-(1.16E-1+7.27E-3*GR-3.64E-6*GR**2)*WS)*100.	RRDST105
	GO TO 100	RRDST106
17	CONTINUE	RRDST107
	GO TO 100	RRDST108
18	TOF=(2.118+1.058E-2*PA+1.014E-4*PA**2)+	RRDST109
•	(2.102E-3+1.84E-4*PA-1.177E-6*PA**2)*T+	RRDST110
•	(1.001E-4-7.046E-7*PA+1.355E-8*PA**2)*T**2	RRDST111
•	GR=(1.0E-5)+(-1.9687+4.209E-1*GW+3.9445E-2*GW**2)*TOF	RRDST112
•	FGR=(GR-(8.363E-2+1.488E-2*GR-9.78E-5*GR**2)*WS)*100.	RRDST113
	GO TO 100	RRDST114
19	TCF=(4.65478+6.94444E-3*T)+(3.257E-1+2.7778E-4*I)*(PA/10.)	RRDST115
•	GR=(.1457+3.5625E-2*GW-6.763E-5*GW**2)+	RRDST116
•	(5.1428-3.175E-2*GW+7.0089E-5*GW**2)*TOF	RRDST117
•	FGR=(GR-(.1+.0082*GR)*WS)*100.	RRDST118
	GO TO 100	RRDST119
20	TOF=(1.2192956+2.2091577E-3*PA+3.380102E-4*PA**2)+	RRDST120
•	(1.4628966E-2+2.6313968E-4*PA-1.3818053E-7*PA**2)*T-	RRDST121
•	(2.4891E-4-6.875E-6*PA+7.8125E-8*PA**2)*T**2+	RRDST122
•	(2.20314E-6-6.49E-8*PA+7.47E-10*PA**2)*T**3	RRDST123



GF= (2.3806396-5.9265772E-2*GW+6.67969E-4*GW**2) +	RRDST124
· (-1.19933136+5.041098E-2*GW-2.12517E-4*GW**2)*TOF)*10.	RRDST125
FGR= (1.0+9.7757143E+1*GR+6.4285714E-2*GR**2) -	RRDST126
· (4.8785706+5.4275515E-1*GR+4.438775E-3*GR**2)*WS	RRDST127
GC TC 100	RRDST128
21 TOF= (-4.799107E-1 + 3.3165178E-2*PA + 2.7902E-4*PA**2) +	RRDST129
· (2.129E-2 + 2.2538E-4 * PA - 2.9186E-6 * PA ** 2) * T	RRDST130
GR = (1.16103 + 5.318E-2 * GW + 9.0525E-4 * GW ** 2) +	RRDST131
· (3.3695E1 - 6.94278E-1 * GW + 3.8559E-3 * GW ** 2) * TOF -	RRDST132
· (-9.041 + 2.307E-1 * GW - 1.264E-3 * GW ** 2) * TOF ** 2 +	RRDST133
· (-1.0708 + 2.477E-2 * GW - 1.108E-4 * GW ** 2) * TOF ** 3	RRDST134
FGR= (GR- (2.4131E-1+2.115E-4*GR + 1.935E-4*GR**2)*WS)*100.	RRDST135
GC TC 100	RRDST136
22 CCNTINUE	RRDST137
23 TOF= (3.9116E-2+6.3976E-2*PA) + (1.6557E-2-7.6643E-6*PA)*T	RRDST138
GR= (5.625-9.5E-2*GW+1.3125E-3*GW**2) +	RRDST139
· (8.6496E-1-1.2768E-2*GW+1.077E-4*GW**2)*TOF+	RRDST140
· (4.0067E-1-5.982E-3*GW+3.627E-5*GW**2)*TOF**2	RRDST141
FGR= (GR- (1.508E-1+8.625E-3*GR)*WS)*100.	RRDST142
GC TC 100	RRDST143
24 TOF= (5.4067E+1-1.3375E-1*PA-2.2755E-4*PA**2+3.6508E-6*PA**3) -	RRDST144
· (7.395E-2-1.71E-4*PA-5.91E-6*PA**2+4.22E-8*PA**3)*T	RRDST145
GR= (8.6549E+3-7.75196E+1*GW+2.07846E-1*GW**2) -	RRDST146
· (5.6302E+2-4.9948*GW+1.30519E-2*GW**2)*TOF+	RRDST147
· (1.22509E+1-1.07805E-1*GW+2.759985E-4*GW**2)*TOF**2-	RRDST148
· (8.8948E-2-7.77463E-4*GW+1.956483E-6*GW**2)*TOF**3	RRDST149
FGR= (GR- (1.4123219E-1+8.5293578E-3*GR+5.709895E-6*GR**2)*WS)*100.	RRDST150
GC TC 100	RRDST151
25 TOF= (7.90371+6.68965E-2*PA+2.12622E-4*PA**2) +	RRDST152
· (3.00808E-2+2.67118E-5*PA+9.85E-6*PA**2)*T+	RRDST153
· (1.23149E-4+1.3589E-6*PA-3.1641E-8*PA**2)*T**2	RRDST154
GR= (2.1742857+2.04286E-1*GW-1.071429E-2*GW**2) +	RRDST155
· (1.14943-1.2707E-1*GW+5.1785E-3*GW**2)*TOF	RRDST156
FGR= (GR- (-2.7327E-2+1.904E-2*GR)*WS+)	RRDST157
· (-6.308077E-4+1.94654E-4*GR)*WS**2)*100.	RRDST158
GC TC 100	RRDST159
26 CCNTINUE	RRDST160
27 CCNTINUE	RRDST161
28 CONTINUE	RRDST162
29 TOF= (7.83935E-1+5.38189E-2*PA) +	RRDST163
· (1.20408E-2+9.888357E-5*PA-2.32448E-6*PA**2)*T-	RRDST164
· (9.72E-6+1.8278E-6*PA-2.405E-8*PA**2)*T**2	RRDST165
GR= (3.18978E+1-1.785*GW+3.602E-2*GW**2) +	RRDST166
· (-8.8285+5.1387E-1*GW-5.679E-3*GW**2)*TOF+	RRDST167
· (-1.76441+4.82709E-2*GW)*TOF**2	RRDST168
FGR= (GR- (8.6457E-2+1.1414E-2*GR)*WS)*100.	RRDST169
GC TC 100	RRDST170
30 TCF= (-2.890514E-1+5.8370956E-2*PA) +	RRDST171
· (4.161561E-2-3.518445E-5*PA)*T+ (-6.0515E-5+3.53095E-6*PA)*T**2	RRDST172
GR= (-2.684337E+1+3.224954*GW) + (-2.0581519+3.7024356E-1*GW)*TOF+	RRDST173
· (-8.861357E-1+8.309318E-2*GW)*TOF**2	RRDST174
FGR= (GR- (1.3583333E-1+9.5833E-3*GR)*WS)*100.	RRDST175
GC TC 100	RRDST176
31 TCF= (7.46275E-1+1.789924E-2*PA+1.667729E-4*PA**2) +	RRDST177
· (6.1017875E-3+3.4816947E-4*PA-1.6406229E-6*PA**2)*T+	RRDST178
· (1.718525E-4-2.621825E-6*PA+4.184375E-8*PA**2)*T**2	RRDST179
GR= (-7.2378129E+1+3.8485684E+1*GW-6.565*GW**2+3.916E-1*GW**3) +	RRDST180
· (-5.477E+1+2.92E+1*GW-4.975*GW**2+2.906E-1*GW**3)*TOF	RRDST181
FGR= ( (-1.607758+1.222176*GR-5.64375E-3*GR**2) -	RRDST182
· (4.82382E-1+2.2260152E-2*GR-4.7462116E-4*GR**2)*WS)*100.	RRDST183
GC TC 100	RRDST184
32 TCF= (1.996+1.69E-2*PA+2.56E-5*PA**2) +	RRDST185



.	(8.64E-3-7.5E-5*PA+1.61E-6*PA**2)*T	FRDST186
.	GR=(6.26E+1-1.299E+1*GW+6.886E-1*GW**2)+	RRDST187
.	(-1.0004E+2+2.0317E+1*GW-9.67E-1*GW**2)*TOF+	RRDST188
.	(1.30368E+1-2.689*GW+1.403E-1*GW**2)*TOF**2	RRDST189
.	FGR=(-3.3E-1+1.047*GR-8.57E-4*GR**2)-	RRDST190
.	(4.22E-2+9.47E-3*GR+1.9898E-5*GR)*WS)*100.	RRDST191
.	GO TC 100	RRDST192
33	TCF=(6.6742857E-1+4.4226786E-2*PA)+	RRDST193
.	(1.027143E-2+3.051339E-4*PA)*T+(1.74994E-4+5.023E-7*PA)*T**2	RRDST194
.	GR=(-1.37666666E+1+1.679166666*GW)+(-3.55+4.71875E-1*GW)*TOF	RRDST195
.	FGR=(GR-(1.516666666E-1+1.008333333E-2*GR)*WS)*100.	RRDST196
.	GO TC 100	RRDST197
34	CCONTINUE	RRDST198
35	CCONTINUE	RRDST199
36	TCF=(-9.2083337E-1+5.9113889E-2*PA)+(2.291666E-2-2.7778E-5*PA)*T	RRDST200
.	GR=(3.711176E+1-1.640279E+1*GW+2.22809*GW**2)+	RRDST201
.	(-2.09922E+1+8.6991796*GW-8.4586E-1*GW**2)*TOF+	RRDST202
.	(2.246949-9.093486E-1*GW+1.061975E-1*GW**2)*TOF**2	RRDST203
.	FGR=(GR-(4.3358E-2+2.196E-2*GR)*WS+	RRDST204
.	(8.79209E-4+8.21219E-5*GR)*WS**2)*100.	RRDST205
.	GC TO 100	RRDST206
37	TCF=(-6.46E-1+6.7857E-2*PA+2.723E-4*PA**2)+	RRDST207
.	(3.69E-2-2.24E-3*PA+3.49E-5*PA**2)*T+	RRDST208
.	(1.07E-4+3.85E-5*PA-4.688E-7*PA**2)*T**2	RRDST209
.	GR=(5.38-1.105*GW+1.14E-1*GW**2)+	RRDST210
.	(8.02E-1-2.57E-1*GW+2.4E-2*GW**2)*TOF	RRDST211
.	FGR=(GR-(1.6E-2+2.44E-2*GR-2.128E-4*GR**2)*WS)*100.	RRDST212
.	GC TC 100	RRDST213
100	RRDIST=FGR	RRDST214
.	RETURN	RRDST215
.	END	RRDST216

FUNCTION SIGY  
(ENTRY: SIGCY)

Purpose:

To compute the horizontal dispersion coefficient in meters, or at entry SIGCY, to compute the virtual distance corresponding to the initial horizontal dispersion.

Input:

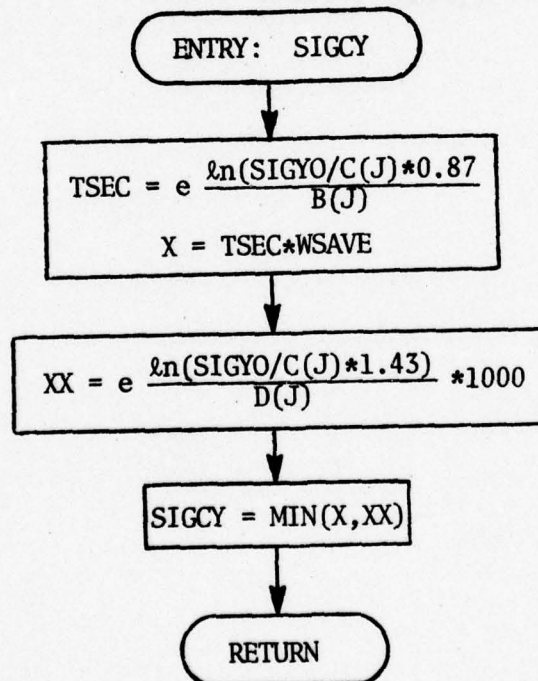
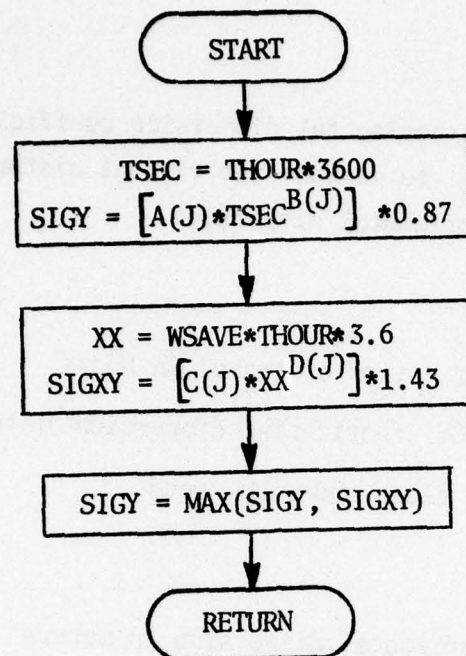
1. Entry SIGY - time of travel in hours
2. Entry SIGCY - horizontal dispersion in meters
3. Stability class and wind speed

Output:

1. SIGY = horizontal dispersion in meters
2. SIGCY = virtual distance in meters

FUNCTION SIGY

ENTRY: SIGCY





	FUNCTION SIGY(J,THOUR)	SIGY0000
C		SIGY0001
C	THIS FUNCTION COMPUTES THE HORIZONTAL DISPERSION COEFFICIENT	SIGY0002
C	IN METERS	SIGY0003
C		SIGY0004
	COMMON /WDUN/ WSAVE	SIGY0005
	DIMENSION A(6),B(6),C(6),D(6)	SIGY0006
	DATA A/2.1511,1.5454,1.0606,.68465,.59366,.59366/	SIGY0007
	DATA B/.87326,.88261,.89031,.88866,.89138,.89138/	SIGY0008
	DATA C /212.,155.,100.,68.,50.,34./	SIGY0009
	DATA D/0.89,0.91,0.92,0.93,0.90,0.93/	SIGY0010
C		SIGY0011
	TSEC=THOUR*3600.	SIGY0012
	SIGY=(A(J)*TSEC**B(J))*0.87	SIGY0013
	XX=WSAVE*THOUR*3.6	SIGY0014
	SIGXY=C(J)*(XX**D(J))*1.43	SIGY0015
	SIGY=AMAX1(SIGY,SIGXY)	SIGY0016
	RETURN	SIGY0017
	ENTRY SIGCY(J,SIGY0)	SIGY0018
C		SIGY0019
C	AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE	SIGY0020
C	INPUT VALUE OF THE HORIZONTAL DISPERSION IS CALCULATED AND	SIGY0021
C	RETURNED AS DISTANCE IN METERS	SIGY0022
C		SIGY0023
	TSEC=EXP(ALOG(SIGY0/(A(J)*0.87)))/B(J)	SIGY0024
	X=TSEC*WSAVE	SIGY0025
	XX=EXP(ALOG(SIGY0/(C(J)*1.43))/D(J))*1000.	SIGY0026
	SIGCY=AMIN1(X,XX)	SIGY0027
	RETURN	SIGY0028
	END	SIGY0029

FUNCTION SIGZ  
(ENTRY: SIGCZ)

Purpose:

To compute the vertical dispersion coefficient in meters, or at entry SIGCZ, to compute the virtual distance corresponding to the initial vertical dispersion.

Input:

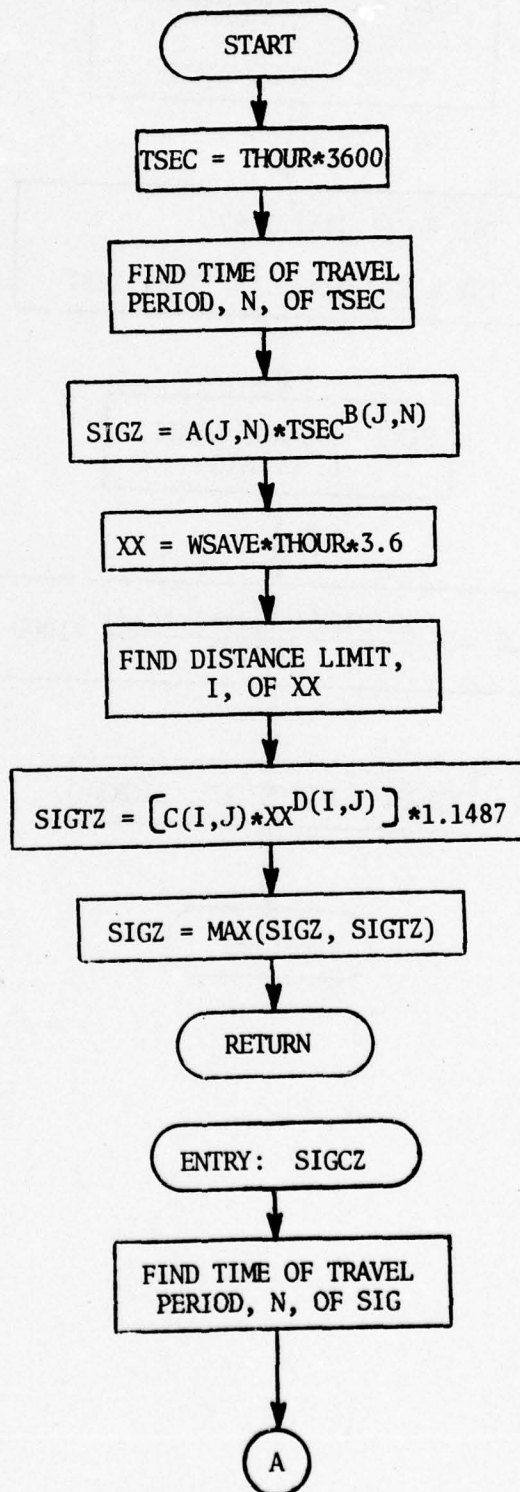
1. Entry SIGZ - time of travel in hours
2. Entry SIGCZ - vertical dispersion in meters
3. Stability class and wind speed

Output:

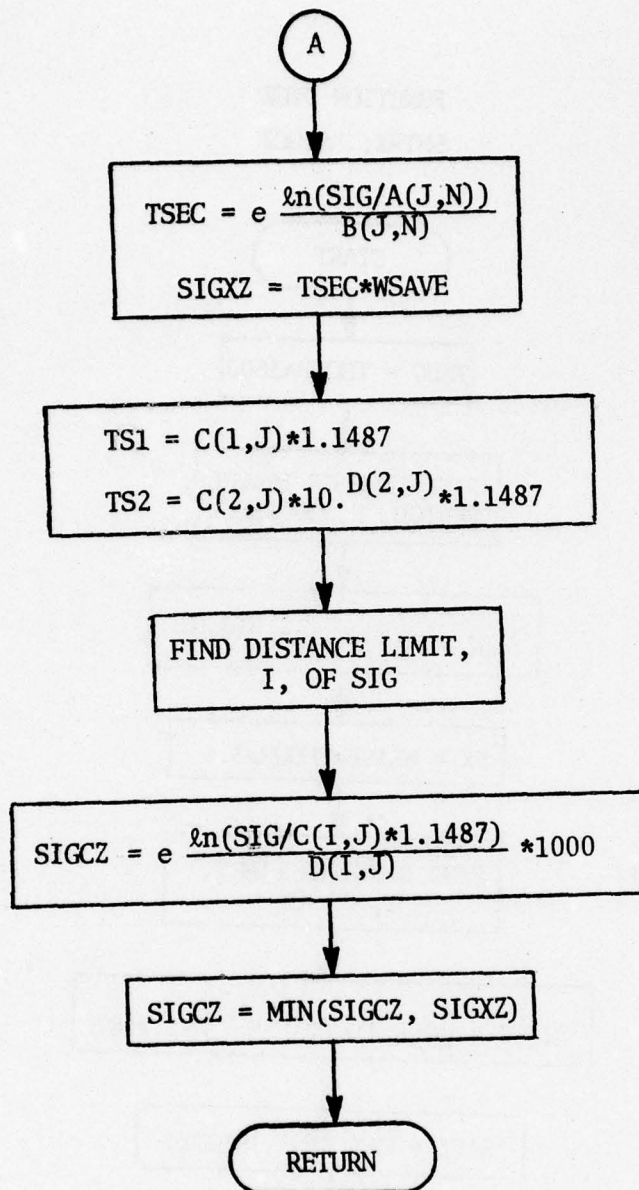
1. SIGZ = vertical dispersion in meters
2. SIGCZ = virtual distance in meters

FUNCTION SIGZ

ENTRY: SIGCZ







C	FUNCTION SIGZ(J,THOUR)	SIGZ0000
C	THIS FUNCTION COMPUTES THE VERTICAL DISPERSION COEFFICIENT	SIGZ0001
C	IN METERS	SIGZ0002
C		SIGZ0003
	COMMON /UDUN/ WSAVE	SIGZ0004
	DIMENSION C(3,6),D(3,6),A(6,6),B(6,6),CK(6,6)	SIGZ0005
	DIMENSION TIME(6)	SIGZ0006
	DATA TIME/ 300.,1000.,3000.,10000.,30000.,172000./	SIGZ0007
	DATA A/.17122,.27668,.41219,.51921,.50963,.47639,	SIGZ0008
1	.11062,.39953,.41219,.57145,.76485,.71936,	SIGZ0009
2	.01338,.16640,.41219,1.0813,1.9467,2.3901,	SIGZ0010
3	.01338,.16640,.41219,2.2830,2.9850,3.8684,	SIGZ0011
3	.01338,.16640,.41219,2.3333,5.7990,16.897,	SIGZ0012
3	.01338,.16640,.41219,5.6801,14.599,64.577/	SIGZ0013
	DATA B/1.2098,1.0572,.92365,.84130,.79689,.76308,	SIGZ0014
1	1.2864,.99275,.92365,.82449,.72571,.69082,	SIGZ0015
2	1.5922,1.1195,.92365,.73217,.59047,.51700,	SIGZ0016
3	1.5922,1.1195,.92365,.63883,.53708,.45686,	SIGZ0017
4	1.5922,1.1195,.92365,.63646,.46497,.29621,	SIGZ0018
5	1.5922,1.1195,.92365,.55016,.37541,.16667/	SIGZ0019
	DATA C/470.,470.,470.,110.,110.,110.,60.,60.,33.,33.,40.,	SIGZ0020
	.21.5,21.5,36.,14.,14.,23.5/	SIGZ0021
	DATA D/1.67,2.13,2.13,1.,1.09,1.09,0.92,0.92,0.92,0.80,0.61,0.53,	SIGZ0022
	.0.70,0.56,0.35,0.78,0.53,0.30/	SIGZ0023
	DATA CK/	SIGZ0024
1	170., 115., 80., 63., 48., 37.,	SIGZ0025
2	800., 380., 243.25, 170., 115., 85.,	SIGZ0026
3	4600., 1300., 671., 380., 220., 150.,	SIGZ0027
4	31279., 5002., 2040.32, 820., 420., 260.,	SIGZ0028
5	179855.2, 17111.38, 5628.47, 1650., 700., 358.,	SIGZ0029
6	2900444., 120872.5, 28241.86, 4312.55, 1348.32, 481.58/	SIGZ0030
C	TSEC=THOUR*3600.	SIGZ0031
	DO 10 N=1,6	SIGZ0032
	IF(TSEC.LE.TIME(N)) GO TO 20	SIGZ0033
10	CONTINUE	SIGZ0034
	N=6	SIGZ0035
C		SIGZ0036
C	TIME OF TRAVEL SHOULD BE LESS THAN 172000 SEC. OR APPROX. 2 DAYS	SIGZ0037
C		SIGZ0038
20	CONTINUE	SIGZ0039
	SIGZ=(A(J,N)*TSEC**B(J,N))	SIGZ0040
	XX=WSAVE*THOUR*3.6	SIGZ0041
	I=1	SIGZ0042
	IF(XX.GT.1.) I=2	SIGZ0043
	IF(XX.GT.10.) I=3	SIGZ0044
C		SIGZ0045
C	CONVERTS FROM A 10 TO 20 MIN. SAMPLING TIME	SIGZ0046
C	1.1487 = 2**.2, THE 1/5 POWER LAW ONLY APPLIES UP TO 20 MIN.	SIGZ0047
C	SAMPLING TIMES	SIGZ0048
C		SIGZ0049
	SIGTZ=(C(I,J)*XX**D(I,J))*1.1487	SIGZ0050
	SIGZ=AMAX1(SIGZ,SIGTZ)	SIGZ0051
	RETURN	SIGZ0052
	ENTRY SIGZ(J,SIG)	SIGZ0053
C		SIGZ0054
C	AT THIS ENTRY THE DISTANCE OR TRAVEL TIME CORRESPONDING TO THE	SIGZ0055
C	INPUT VALUE OF THE VERTICAL DISPERSION IS CALCULATED AND	SIGZ0056
C	RETURNED AS DISTANCE IN METERS	SIGZ0057
C		SIGZ0058
	DO 110 N=1,6	SIGZ0059
		SIGZ0060
		SIGZ0061

```

      IF (SIG.LE.CK(J,N)) GO TO 120
110  CONTINUE
      N=6
120  CONTINUE
      TSEC=EXP(ALOG(SIG/A(J,N))/B(J,N))
      SIGXZ=TSEC*WSAVE
      TS1=C(1,J)*1.1487
      TS2=C(2,J)*10.**D(2,J)*1.1487
      I=3
      IF (SIG.LT.TS2) I=2
      IF (SIG.LT.TS1) I=1
      SIGCZ=EXP(ALOG(SIG/(C(I,J)*1.1487))/D(I,J))*1000.
      SIGCZ=AMIN1(SIGCZ,SIGXZ)
      RFTURN
      END

```

```

SIGZ0062
SIGZ0063
SIGZ0064
SIGZ0065
SIGZ0066
SIGZ0067
SIGZ0068
SIGZ0069
SIGZ0070
SIGZ0071
SIGZ0072
SIGZ0073
SIGZ0074
SIGZ0075
SIGZ0076

```



## SUBROUTINE SOURCE

### Purpose:

To position the master source tape to read the airbase and environ source inventory data and to call the subroutines which compute the emission rates in micrograms per second at the airbase and environ sources.

### Input:

JFLAG, a parameter to indicate whether the diurnal distribution used is input, default or the same as previous hour.

### Output:

A statement indicating the diurnal distribution used.

### Subroutines

#### Called:

ABPTAR, ABARAR, ABLNAR, ENARAY

C	SUBROUTINE SOURCE	SOURC001
C	THIS ROUTINE SERVES AS A DRIVER TO CALL SUBROUTINES	SOURC002
C	WHICH COMPUTE THE EMISSION RATES IN MICROGRAMS	SOURC003
C	PER SECOND AT THE AIRBASE AND ENVIRON SOURCES	SOURC004
C		SOURC005
	COMMON / DEFALT / ITAPE	SOURC006
	COMMON /PERIOD/ IMONTH,NODAYS, IDAY, IHR1, IHR2, IFLAG, JFLAG, IONCE	SOURC007
	COMMON/JUNK/DAYS,LSRCE,NSRCE,SORCE(17,300),SORGM(10,200)	SOURC008
	,LCC1,LOC2,NGEOM,IPT	SOURC009
	DIMENSION NAME(2)	SOURC010
	DATA NAME /4H1/12,4H1 /	SOURC011
	IF (IONCE.EQ.0) GO TO 30	SOURC012
	IEND=1	SOURC013
	IST=1	SOURC014
	GC TO 40	SOURC015
30	IEND=0	SOURC016
	IONCE=1	SOURC017
	IST=0	SOURC018
40	CCONTINUE	SOURC019
	DAYS=NODAYS	SOURC020
	IF (IST.EQ.1) GO TO 3	SOURC021
1	FORMAT (I4)	SOURC022
	READ 1, JFLAG	SOURC023
	IF (JFLAG) 8,7,3	SOURC024
7	FFINT 5	SOURC025
5	FORMAT(32HGINPUT DIURNAL DISTRIBUTION USED)	SOURC026
	GC TO 4	SOURC027
8	I=1	SOURC028
	IF (NODAYS.EQ.365) I=2	SOURC029
	PRINT 9,NAME(I)	SOURC030
9	FORMAT(34HDEFAULT DIURNAL DISTRIBUTION USED/5X,12HHOUR = 1/24,5X,	SOURC031
	.10HLAY = 1/7,5X,8HMONTH = A4,1H,5X,12HUNIFAC = 0.1)	SOURC032
	GC TO 4	SOURC033
3	PRINT 6	SOURC034
6	FORMAT(39HDIURNAL DISTRIBUTION SAME AS LAST HOUR)	SOURC035
	GC TO 10	SOURC036
4	IF (IEND.EQ.0) GO TO 12	SOURC037
11	READ (ITAPE,END=12)	SOURC038
	GC TO 11	SOURC039
12	IEND=1	SOURC040
	CALL ABETAR	SOURC041
	CALL ABAARAR	SOURC042
	CALL ABLNAR	SOURC043
	CALL ENARAY	SOURC044
10	REWIND ITAPE	SOURC045
	RETURN	SOURC046
	END	SOURC047
		SOURC048

## SUBROUTINE STPOL1

### Purpose:

To calculate pollutant concentrations from point and area sources.

### Input:

1. Location and conditions at point or area source
2. Location of receptors
3. Meteorological conditions

### Output:

Concentration of pollutants at each receptor

### Procedure:

1. For area sources determine average diameter, effective stack height, and initial values of horizontal and vertical dispersion. Also consider effects of downwash on these.
2. For point sources determine plume rise by calling PLRISE.
3. Consider effects of wind speed at height of source.
4. Calculate crosswind and downwind components for the source.
5. Calculate time required for plume to travel from virtual point source to actual location of true source by PSEUDO routine.
6. For each receptor, calculate crosswind and downwind components.
7. Consider the relative location of receptor with respect to source and, if necessary, calculate coupling coefficient using TRAN routine.
8. For each pollutant, add in concentration determined.

### Subroutines Called:

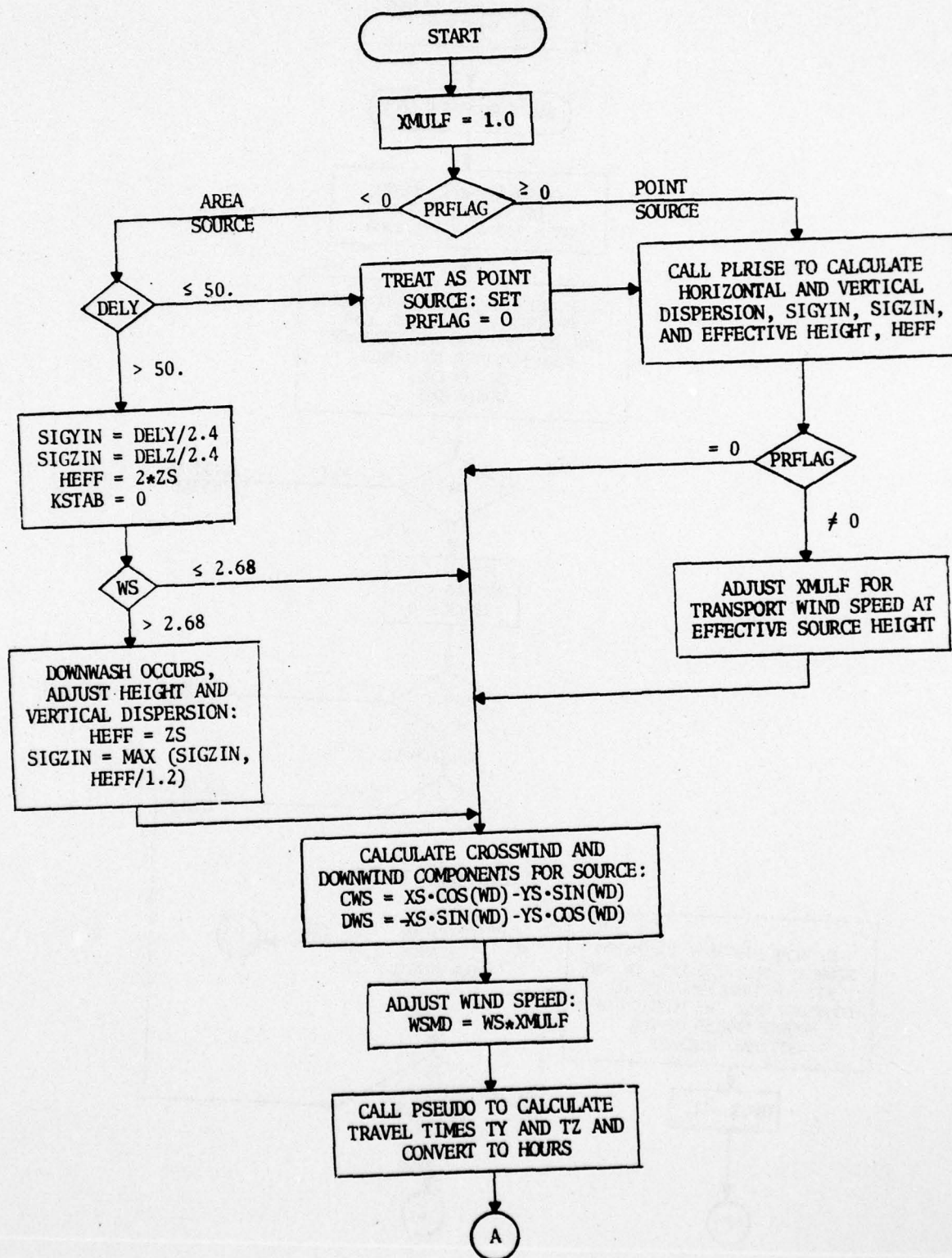
PLRISE, PSEUDO

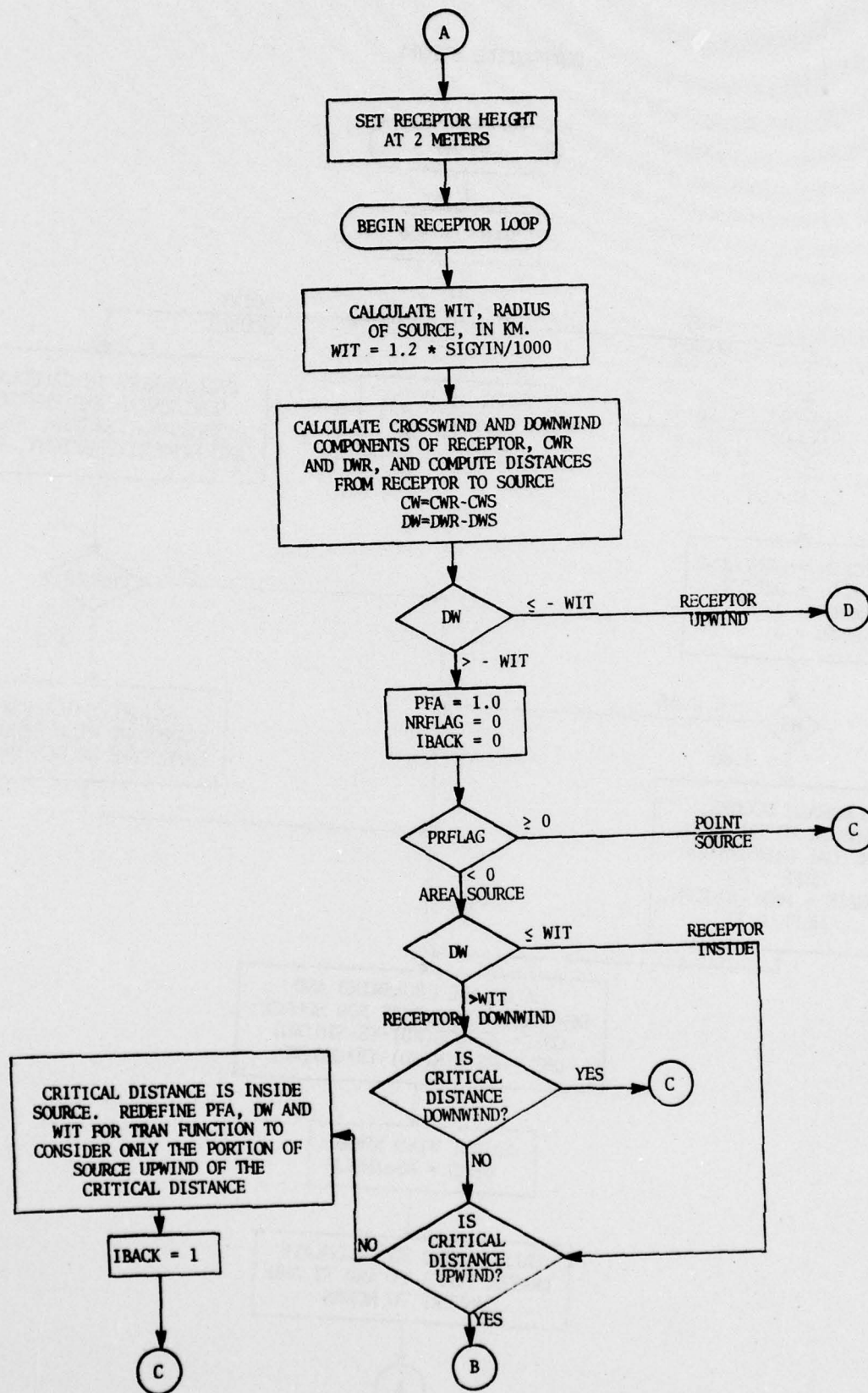
### Function Called:

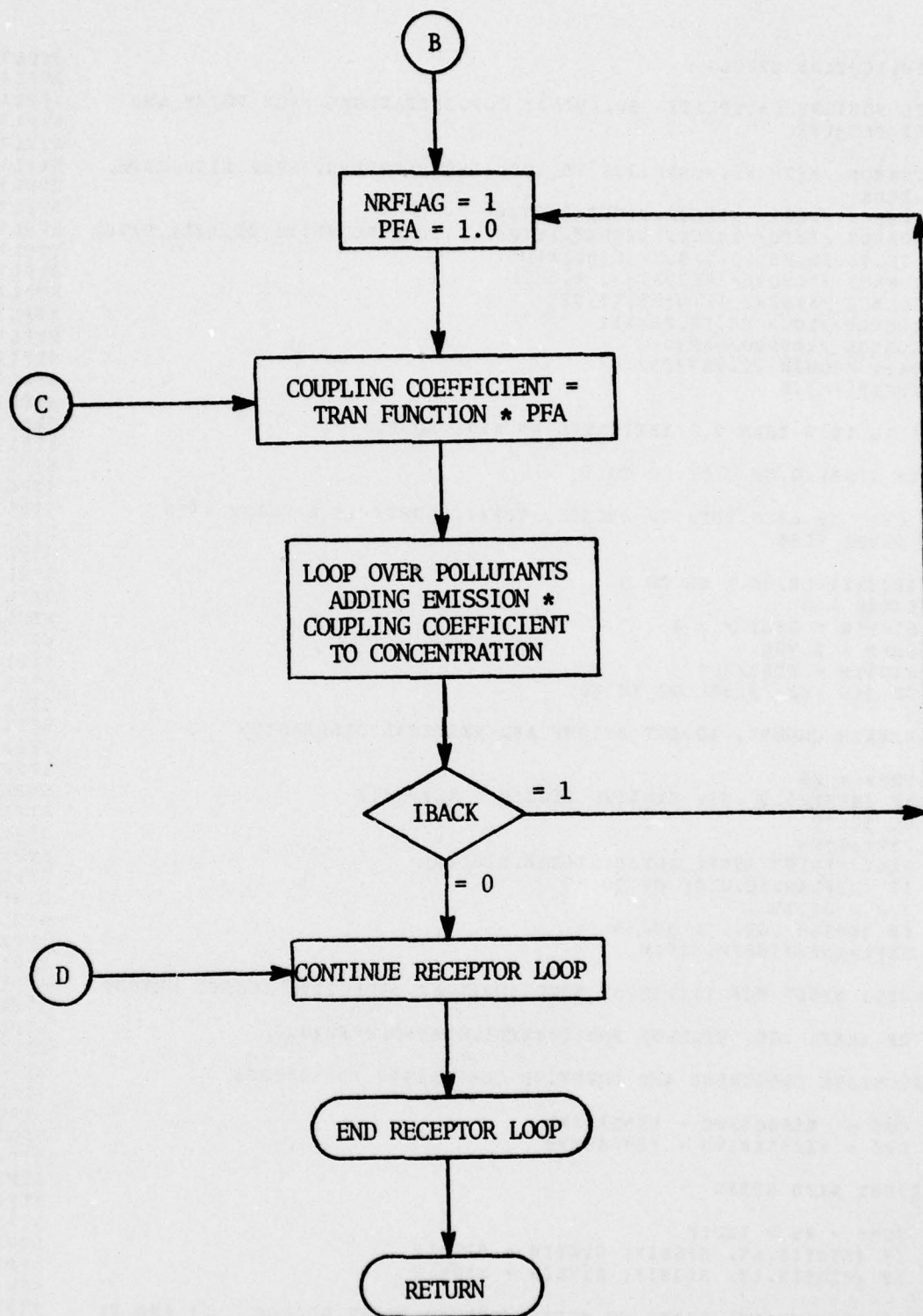
TRAN



# SUBROUTINE STPOL1









C	SUBROUTINE STPOL1	STPL1000
C	THIS ROUTINE CALCULATES POLLUTANT CONCENTRATIONS FROM POINT AND	STPL1001
C	AREA SOURCES	STPL1002
C		STPL1003
	COMMON /NET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HLID,TEMP,	STPL1004
	. TENK	STPL1005
	COMMON /RCPT/ NRECEP,RECEP(2,312)	STPL1006
	COMMON /INFO/ IRECEP,IWDIR,ITYPE ,HTAERO,XS,YS,ZS,DELY,DELZ,	STPL1007
	. IS,VS,DS,HB,PRFLAG,EMIS(8),NPOL	STPL1008
	COMMON /AIRQAL/ RECDAT(3, 6,312)	STPL1009
	COMMON /XTRAN/ XL,WSMD,TY,TZ	STPL1010
	COMMON /LOC/ DW,CW,ZR,WIT	STPL1011
	COMMON /WMDPRO/ XP(6)	STPL1012
	DATA SIGMIN /2.083333/	STPL1013
	XHULF = 1.0	STPL1014
C		STPL1015
C	PRFLAG LESS THAN 0.0 INDICATES AN AREA SOURCE	STPL1016
C		STPL1017
C	IF (PRFLAG.GE.0.0) GO TO 5	STPL1018
C		STPL1019
C	IF DELY IS LESS THAN 50 METERS, TREAT SOURCE AS A POINT WITH	STPL1020
C	NO PLUME RISE	STPL1021
C		STPL1022
	IF(DELY.LE.50.) GO TO 4	STPL1023
	KSTAB = 0	STPL1024
	SIGYIN = DELY / 2.4	STPL1025
	HEFF = 2.*ZS	STPL1026
	SIGZIN = DELZ/2.4	STPL1027
	IF (WS .LE. 2.68) GO TO 10	STPL1028
C		STPL1029
C	DOWNWASH OCCURS, ADJUST HEIGHT AND VERTICAL DISPERSION	STPL1030
C		STPL1031
	HEFF = ZS	STPL1032
	IF (HEFF/1.2 .GT. SIGZIN) SIGZIN = HEFF/1.2	STPL1033
	GO TO 10	STPL1034
	4 PRFLAG=0.	STPL1035
	5 CALL FLRISE (HEFF,KSTAB,SIGZIN,SIGYIN)	STPL1036
	IF (PRFLAG.EQ.0.0) GO TO 10	STPL1037
	JJJ = JSTAB	STPL1038
	IF (KSTAB .GE. 1) JJJ=5	STPL1039
	HEFL=AMIN1(HEFF,305.)	STPL1040
C		STPL1041
C	ADJUST XHULF FOR TRANSPORT WIND SPEED AT EFFECTIVE SOURCE HEIGHT	STPL1042
C		STPL1043
C	IF (HEFL .GT. HTAERO) XHULF=(HEFL/HTAERO)**XP(JJJ)	STPL1044
C		STPL1045
C	CALCULATE CROSSWIND AND DOWNWIND COMPONENTS FOR SOURCE	STPL1046
C		STPL1047
	10 CWS = XS*COSEWD - YS*SINEWD	STPL1048
	DWS = -XS*SINEWD - YS*COSEWD	STPL1049
C		STPL1050
C	ADJUST WIND SPEED	STPL1051
C		STPL1052
	WSMD = WS * XHULF	STPL1053
	IF (SIGYIN.LT. SIGMIN) SIGYIN = SIGMIN	STPL1054
	IF (SIGZIN.LT. SIGMIN) SIGZIN = SIGMIN	STPL1055
C		STPL1056
C	CALCULATE TRAVEL TIMES TO PSEUDO UPWIND POINT SOURCE. TY AND TZ	STPL1057
C	ARE RETURNED AS DISTANCES IN METERS AND CONVERTED TO EQUIVALENT	STPL1058
C	TIMES IN HOURS	STPL1059
C		STPL1060
		STPL1061

CALL PSEUDO(SIGYIN,WSMD,SIGZIN,TY,TZ)	STPL1062
TY=TY/WSMD/3600.	STPL1063
TZ=TZ/WSMD/3600.	STPL1064
ZR = 2.	STPL1065
C BEGIN RECEPTOR LOOP	STPL1066
C	STPL1067
C DC 20 NR=1,NRECEP	STPL1068
C	STPL1069
C WIT IS RADIUS OF SOURCE IN KILOMETERS	STPL1070
C	STPL1071
C WIT = 1.2 * SIGYIN / 1000.	STPL1072
C	STPL1073
C CALCULATE CROSSWIND AND DOWNWIND COMPONENTS OF RECEPTOR AND	STPL1074
C COMPUTE DISTANCES IN KILOMETERS FROM RECEPTOR TO SOURCE	STPL1075
C	STPL1076
C CWR = RECEP(1,NR)*COSEWD - RECEP(2,NR)*SINEWD	STPL1077
C CW = CWR - CWS	STPL1078
C DWR = -RECEP(1,NR) * SINEWD - RECEP(2,NR) * COSEWD	STPL1079
C DW = DWR - DWS	STPL1080
C	STPL1081
C PFA IS FFACTION OF TOTAL AREA SOURCE BEING TREATED	STPL1082
C	STPL1083
C IF (DW.LE.-WIT) GO TO 20	STPL1084
C	STPL1085
C IS RECEPTOR UPWIND OF SOURCE?	STPL1086
C	STPL1087
C	STPL1088
C PFA = 1.0	STPL1089
C NRFLAG=0	STPL1090
C IBACK=0	STPL1091
C IF (PRFLAG.GE.0.0) GO TO 16	STPL1092
C	STPL1093
C IS RECEPTOR INSIDE SOURCE?	STPL1094
C	STPL1095
C IF(DW.LE.WIT) GO TO 15	STPL1096
C	STPL1097
C IS CRITICAL DISTANCE DOWNWIND OF DOWNWIND EDGE OF SOURCE?	STPL1098
C	STPL1099
C IF ((DWR-XL) .GE. (DWS+WIT)) GO TO 16	STPL1100
C	STPL1101
C IS CRITICAL DISTANCE UPWIND OF UPWIND EDGE OF SOURCE?	STPL1102
C	STPL1103
C 15 IF ((DWR-XL) .LT. (DWS-WIT)) GO TO 17	STPL1104
C	STPL1105
C CRITICAL DISTANCE IS INSIDE SOURCE. REDEFINE PFA, DW AND WIT	STPL1106
C PCR TRAN FUNCTION TO CONSIDER ONLY THE PORTION OF SOURCE	STPL1107
C UPWIND OF THE CRITICAL DISTANCE	STPL1108
C	STPL1109
C PFA=(DW-XL+WIT)/(2.*WIT)	STPL1110
C DWSS=((DWS-WIT)+(DWR-XL))/2.	STPL1111
C DW=DWR-DWSS	STPL1112
C WIT=DW-XL	STPL1113
C IBACK=1	STPL1114
C GO TO 16	STPL1115
C 17 NRFLAG=1	STPL1116
C PFA=1.	STPL1117
C	STPL1118
C CALL TRAN FUNCTION TO DETERMINE COUPLING COEFFICIENTS	STPL1119
C	STPL1120
C 16 CUPCOE = TRAN (KSTAB,HEFF,NRFLAG,IBACK) * PFA	STPL1121
C	STPL1122
C ADD EMISSIONS TIMES COUPLING COEFFICIENT TO CONCENTRATIONS	STPL1123

C AT ALL RECEPTORS  
C

DO 18 IPOL = 1, NPOL  
18 RECDAT(ITYPE, IPOL, NR) = RECDAT(ITYPE, IPOL, NR) + EMIS(IPOL) \* CUPCOESTPL1127  
IF(IBACK.EQ.1) GO TO 17  
20 CONTINUE  
RETURN  
END

STPL1124  
STPL1125  
STPL1126  
STPL1127  
STPL1128  
STPL1129  
STPL1130  
STPL1131



## SUBROUTINE STPOL2

### Purpose:

To prepare the data required by the line source model and, for each receptor, to call the model and then add the pollutant concentrations calculated to the accumulated totals at that receptor.

### Input:

1. Source parameters for the current line.
2. Wind speed and direction, and lid height.

### Output:

Accumulated pollutant concentrations at all receptors.

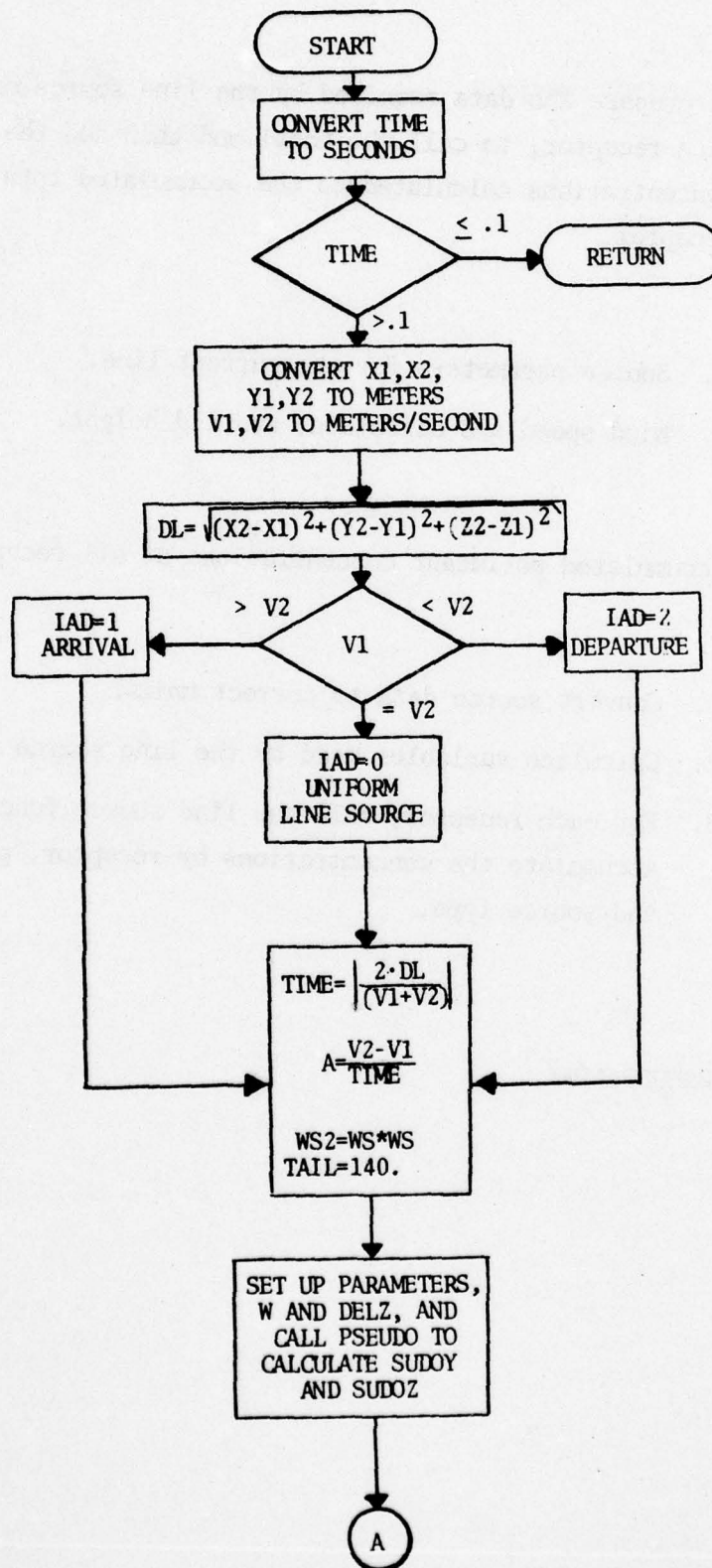
### Procedure:

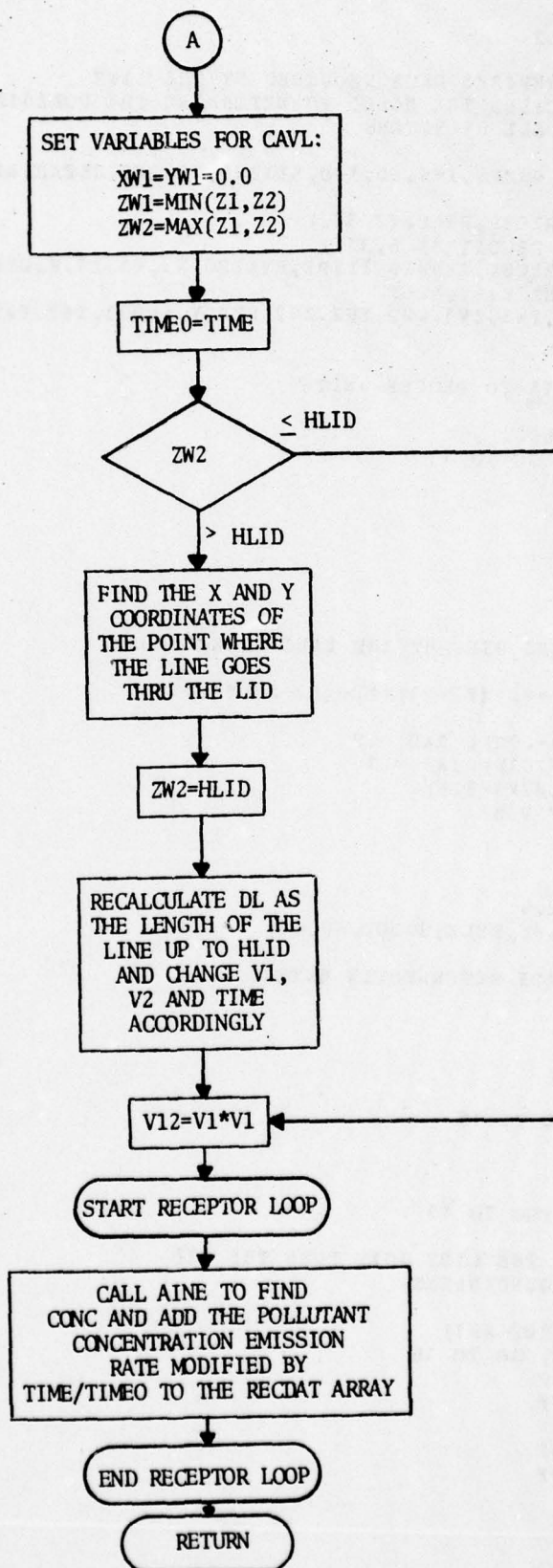
1. Convert source data to correct units.
2. Calculate variables used by the line source model.
3. For each receptor, call the line source function and accumulate the concentrations by receptor, pollutant, and source type.

### Subroutines Called:

PSEUDO,AINE

SUBROUTINE STPOL2







AD-A046 348

ARGONNE NATIONAL LAB ILL  
AIR QUALITY ASSESSMENT MODEL FOR AIR FORCE OPERATIONS - SHORT-T--ETC(U)  
APR 77 D J BINGAMAN

F/G 13/2

UNCLASSIFIED

CEEDO-TR-76-34

NL

3 OF 3

AD  
A046348



END  
DATE  
FILMED  
12-77  
DDC

	SUBROUTINE STPOL2	STPL2000
C		STPL2001
C	THIS SUBROUTINE PREPARES DATA REQUIRED BY THE LINE	STPL2002
C	SOURCE MODEL AND CALLS THE MODEL TO DETERMINE THE POLLUTANT	STPL2003
C	CONCENTRATIONS AT ALL RECEPTORS	STPL2004
C		STPL2005
	COMMON /MET/ WS, WSHPH, IWS, WD, IWD, SINEWD, COSEWD, JSTAB, HLID, TEMP,	STPL2006
	. IENK	STPL2007
	CCHMCN /RCPT/ NRECEP, RECEP(2, 312)	STPL2008
	COMMON /AIRQAL/ RECDAT(3, 6, 312)	STPL2009
	COMMON /INFO/ IRECEP, IWDIR, ITYPE, HTAERO, X1, Y1, Z1, W, DELZ, X2, Y2, Z2,	STPL2010
	. V1, V2, DL, TIME, EMIS(6), NPOL	STPL2011
	COMMON /LM/ XW1, YW1, ZW1, XW2, YW2, ZW2, SUDOY, SUDOZ, IAD, TAIL, A, V12, VS,	STPL2012
	. WS2, WSC, RR, SP	STPL2013
C		STPL2014
C	CONVERT SOURCE DATA TO PROPER UNITS	STPL2015
C		STPL2016
	TIME = TIME * 3600.	STPL2017
	IF(TIME.LE.0.1) GO TO 11	STPL2018
	X1 = X1 * 1000.	STPL2019
	Y1 = Y1 * 1000.	STPL2020
	X2 = X2 * 1000.	STPL2021
	Y2 = Y2 * 1000.	STPL2022
	V1 = V1 / 3.6	STPL2023
	V2 = V2 / 3.6	STPL2024
C		STPL2025
C	CALCULATE VARIABLES USED BY THE LINE SOURCE MODEL	STPL2026
C		STPL2027
	DL=SQRT((X2-X1)**2+(Y2-Y1)**2+(Z2-Z1)**2)	STPL2028
	IAD = 0	STPL2029
	IF (V1 .LT. (V2-.01)) IAD = 2	STPL2030
	IF (V1 .GT. (V2+.01)) IAD = 1	STPL2031
	TIME = ABS(2*DL/(V1+V2))	STPL2032
	A = (V2 - V1) / TIME	STPL2033
	WS2 = WS * WS	STPL2034
	TAIL = 140.	STPL2035
	W = W / 2.4	STPL2036
	DELZ = DELZ / 2.4	STPL2037
	CALL PSEUDO (W, WS, DELZ, SUDOY, SUDOZ)	STPL2038
C		STPL2039
C	SUDOY AND SUDOZ ARE RETURNED IN METERS	STPL2040
C		STPL2041
	XW1=0.	STPL2042
	YW1=0.	STPL2043
	ZW1=Z1	STPL2044
	ZW2=Z2	STPL2045
	IF(Z1.LE.Z2) GO TO 15	STPL2046
	ZW1=Z2	STPL2047
	ZW2=Z1	STPL2048
15	TIMEO=TIME	STPL2049
	IF(ZW2.LE.HLID) GO TO 18	STPL2050
C		STPL2051
C	FIND POINT WHERE THE LINE GOES THRU THE LID	STPL2052
C	AND CHANGE THE COORDINATES	STPL2053
C		STPL2054
	F=(HLID-ZW1)/(ZW2-ZW1)	STPL2055
	IF (Z1 .GT. Z2) GO TO 16	STPL2056
	X2=X1+(X2-X1)*F	STPL2057
	Y2=Y1+(Y2-Y1)*F	STPL2058
	GC TO 17	STPL2059
16	X1=X2+(X1-X2)*F	STPL2060
	Y1=Y2+(Y1-Y2)*F	STPL2061

```

17 ZW2 = HLID
C
C RECALCULATE THE LENGTH OF THE LINE UP TO HLID AND
C CHANGE VELOCITIES ACCORDINGLY
C
DLSQ = (X1-X2)**2 + (Y1-Y2)**2 + (ZW1-ZW2)**2
DL = SQRT(DLSQ)
IF (Z2 .GT. Z1) V2 = SQRT(V1*V1+2.*A*DL)
IF (Z2 .LT. Z1) V1 = SQRT(V2*V2-2.*A*DL)
TIME = 2* DL / (V1+V2)
18 V12 = V1 * V1
C
C CALL THE LINE FUNCTION TO DETERMINE POLLUTANT CONCENTRATIONS
C AT ALL RECEPTORS
C
DO 10 IRECEP=1,NRECEP
CONC = AINE(WD)
DO 10 IPOL=1,NPOL
10 RECDAT (ITYPE,IPOL,IRECEP) = RECDAT(ITYPE,IPOL,IRECEP) +
- EMIS(IPOL) * CONC * TIME / TIMEO
11 CCNTINUE
RETURN
END

```

```

STPL2062
STPL2063
STPL2064
STPL2065
STPL2066
STPL2067
STPL2068
STPL2069
STPL2070
STPL2071
STPL2072
STPL2073
STPL2074
STPL2075
STPL2076
STPL2077
STPL2078
STPL2079
STPL2080
STPL2081
STPL2082
STPL2083
STPL2084

```



## FUNCTION TRAN

### Purpose:

To calculate the coupling coefficient at a receptor due to a point or area source.

### Input:

1. Meteorological conditions: wind speed, stability, mixing height, critical distance for mixing.
2. Source parameters: initial horizontal and vertical dispersion; effective stack height; pseudo transport times corresponding to the dispersions; plume height flag, KSTAB; area source flags, NRFLAG and IBACK.

### Output:

Point or area source coupling coefficient, TRAN.

### Procedure:

1. If the effective stack height exceeds the mixing height, then the stability index is reassigned according to the KSTAB flag, the lid is set at 3050 meters and the critical distance at 100 meters.
2. For sources with NRFLAG=0, compute the travel time for z dispersion from the center and that for y dispersion from the downwind edge of the source. Then the effects of ground and sky lid are treated by the image method, with up to 6 terms included in the coupling coefficient.
3. For area sources with NRFLAG=1, the travel times from the upwind and downwind edges of the source are determined on the basis of receptor location relative to the source. These plus the pseudo travel time, TZ, due to the z spread are used to compute the z-dispersion coefficients  $\sigma_z$  (T1) and  $\sigma_z$  (T2). The y-dispersion coefficient  $\sigma_y$  (TT) is determined on the basis of the pseudo travel time, TY, due to the y-spread plus the travel time from the downwind edge

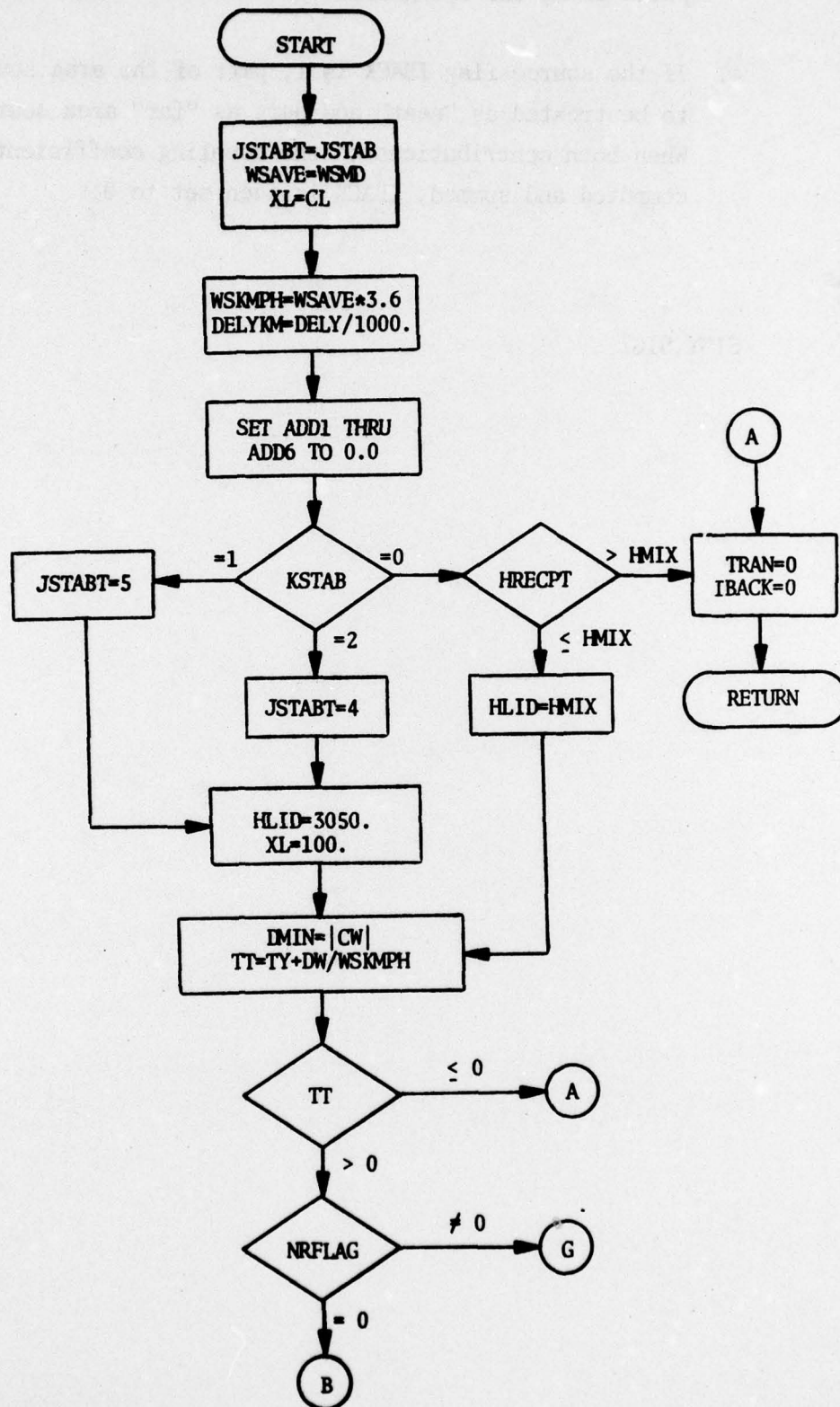
to the receptor. Then the coupling coefficient is computed using the integrated expression for "near" source.

4. If the source flag IBACK is 1, part of the area source is to be treated as "near" and part as "far" area sources. When both contributions to the coupling coefficient are computed and summed, IBACK is then set to 0.

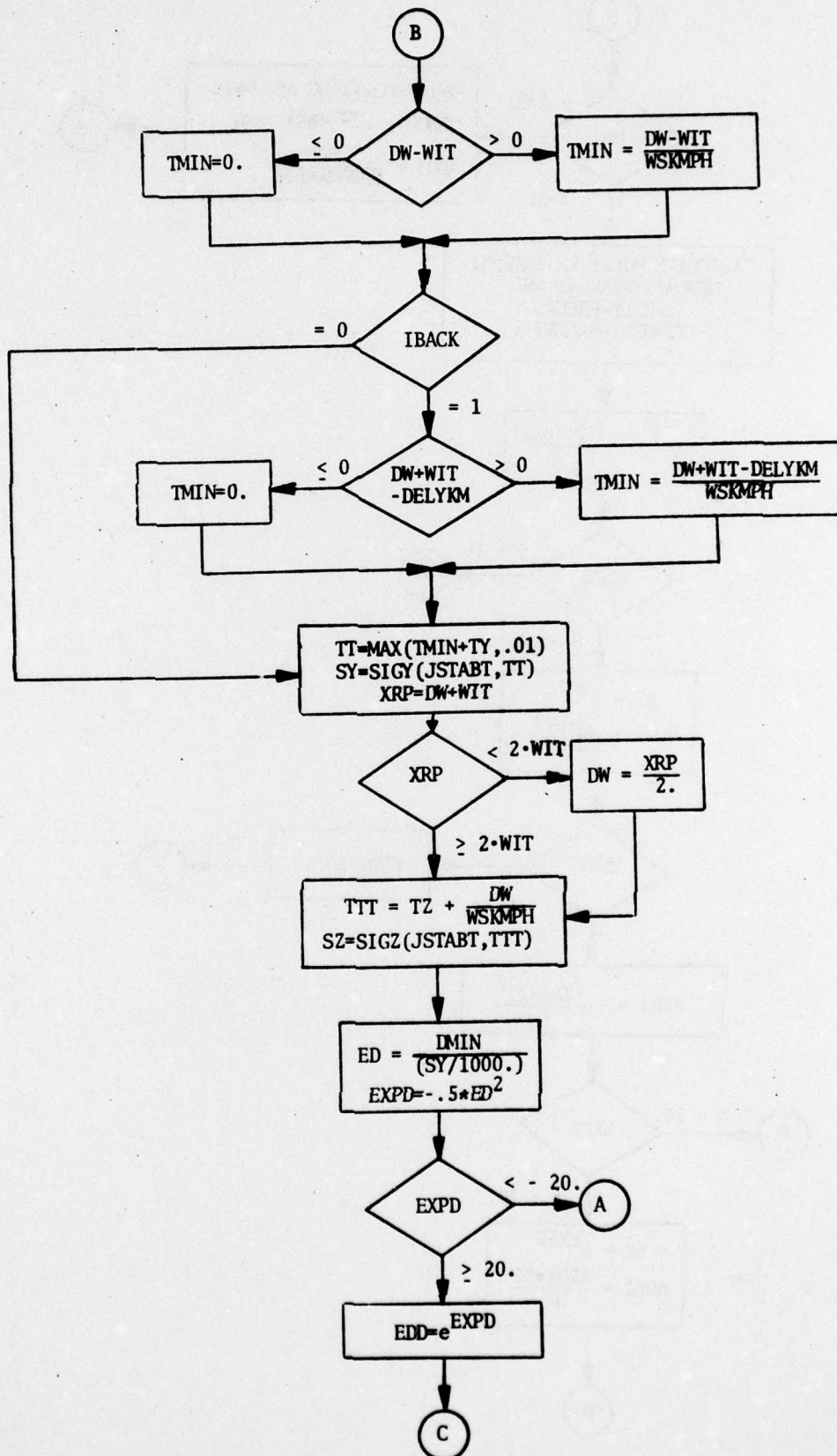
Functions  
Called:

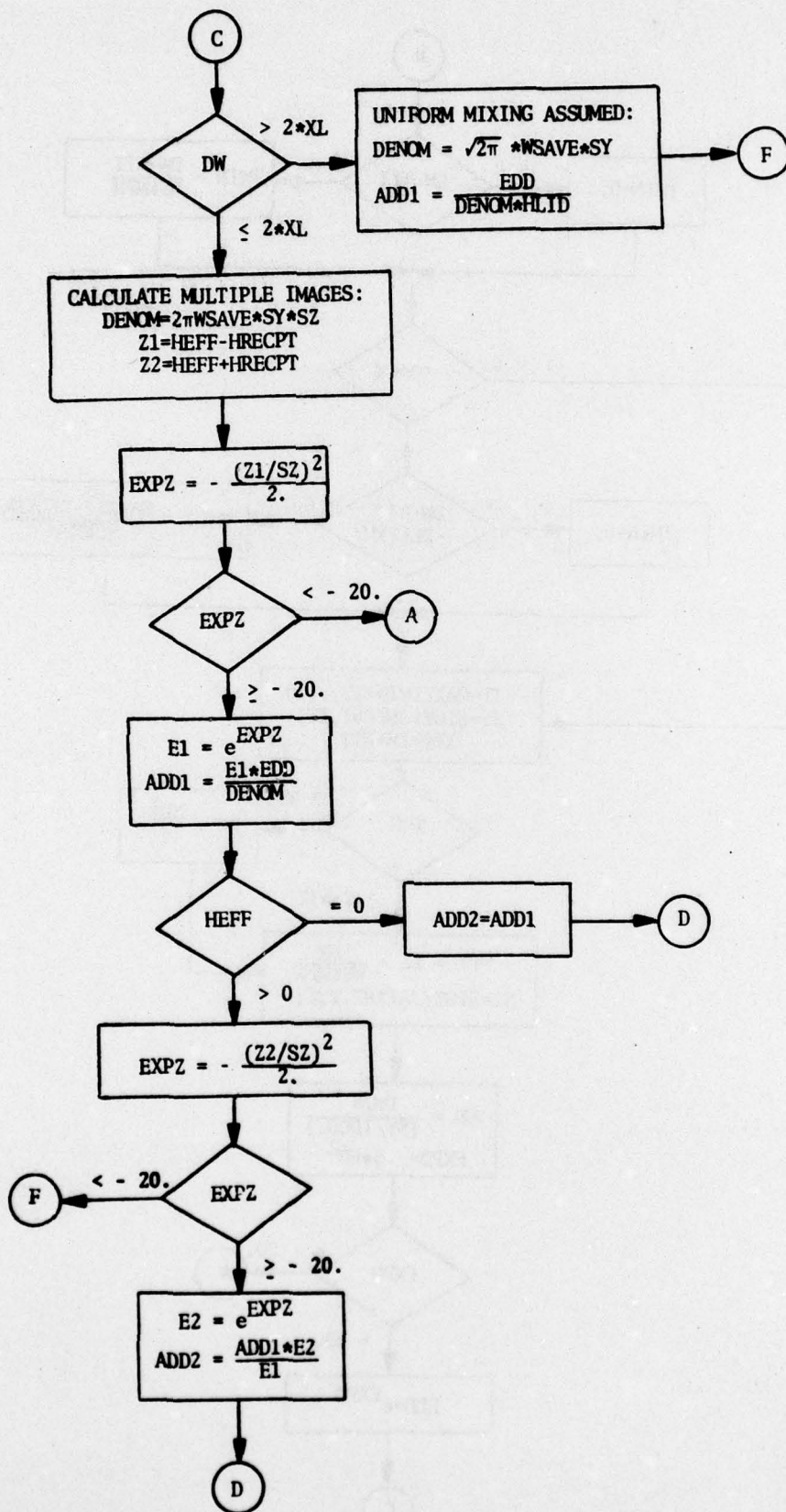
SIGY, SIGZ

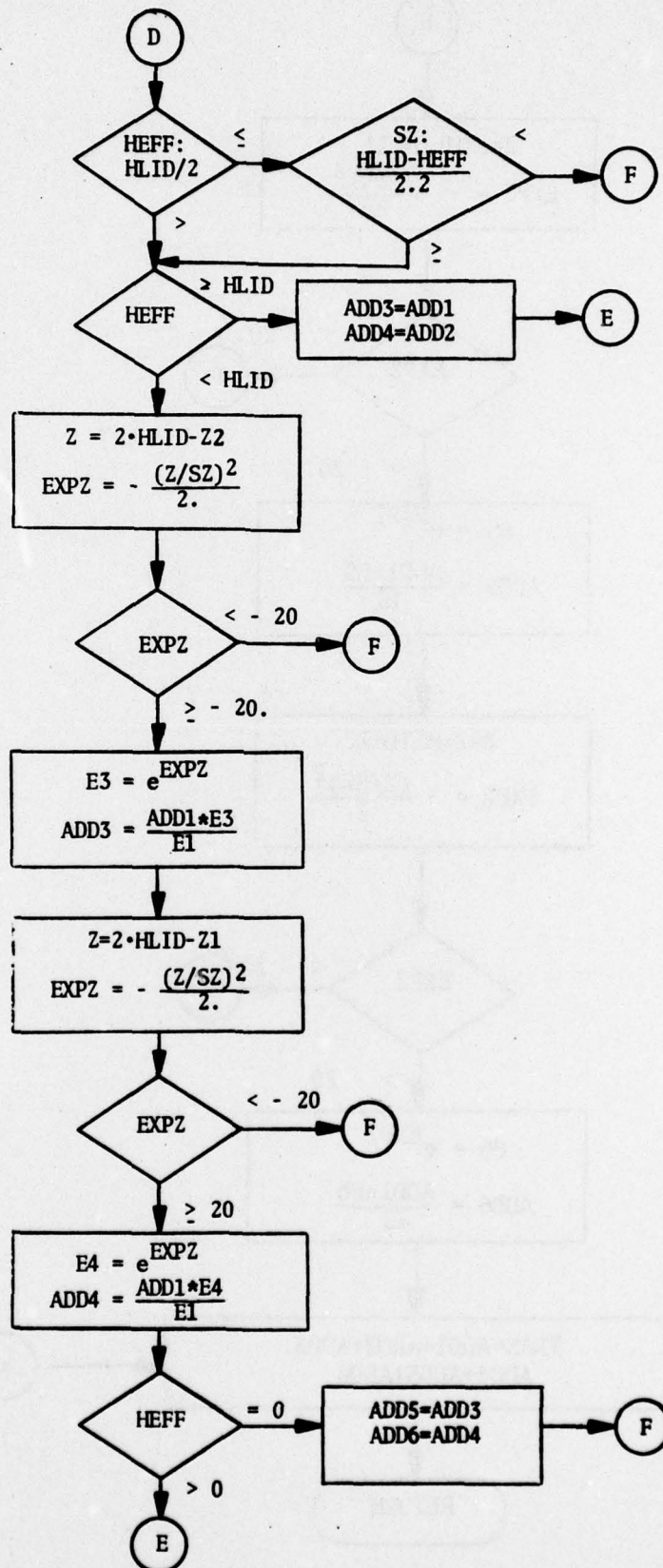
FUNCTION TRAN (KSTAB, HEFF, NRFLAG, IBACK)



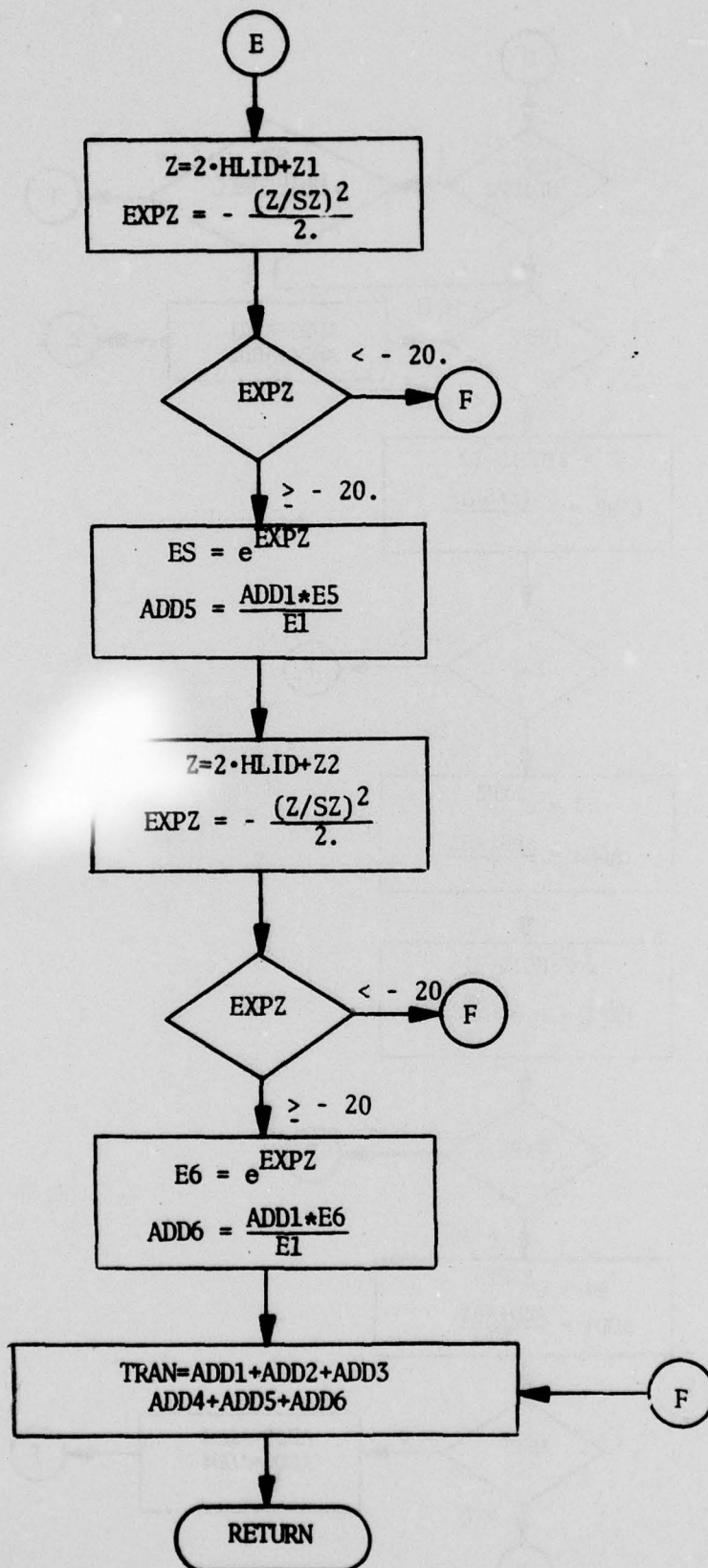


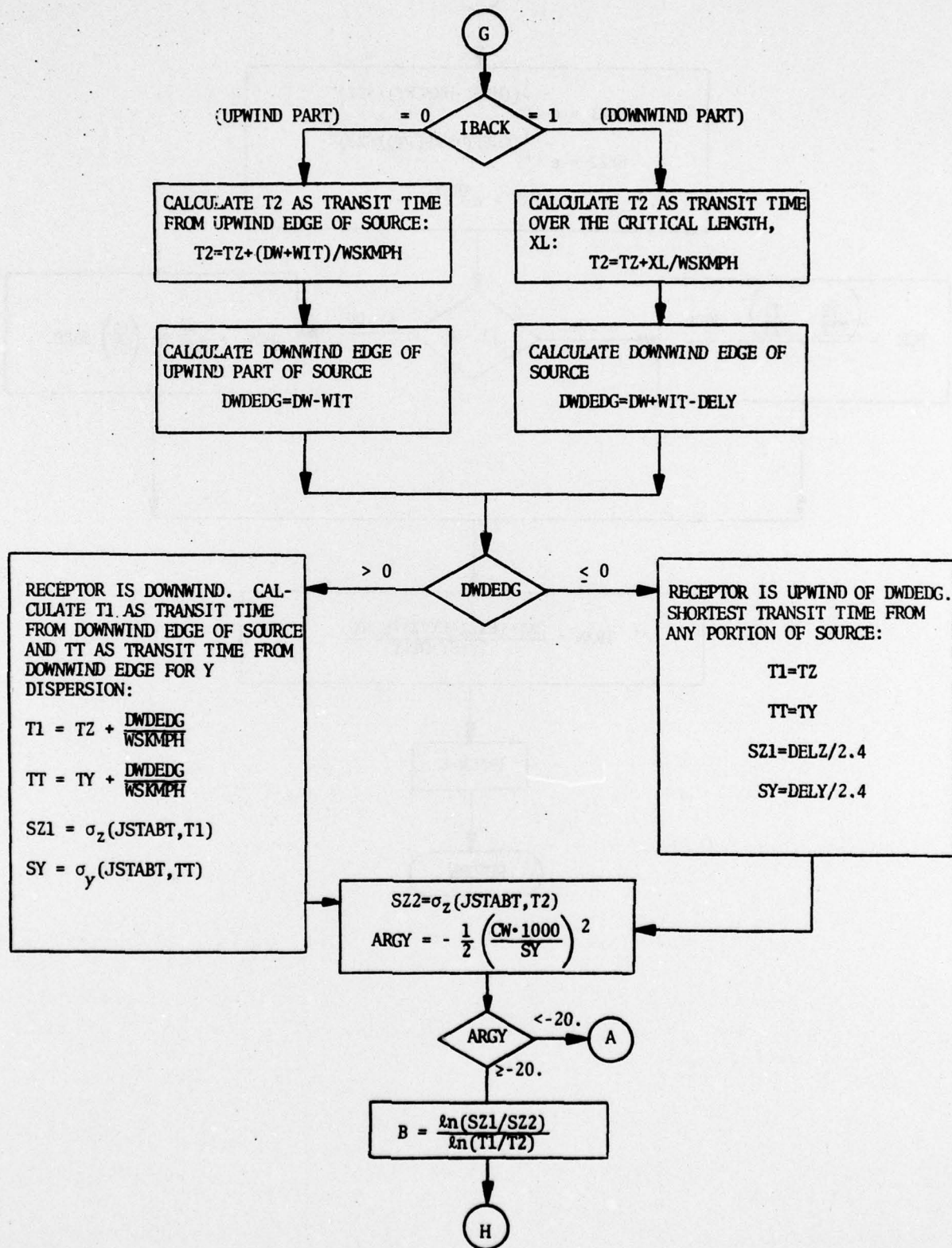


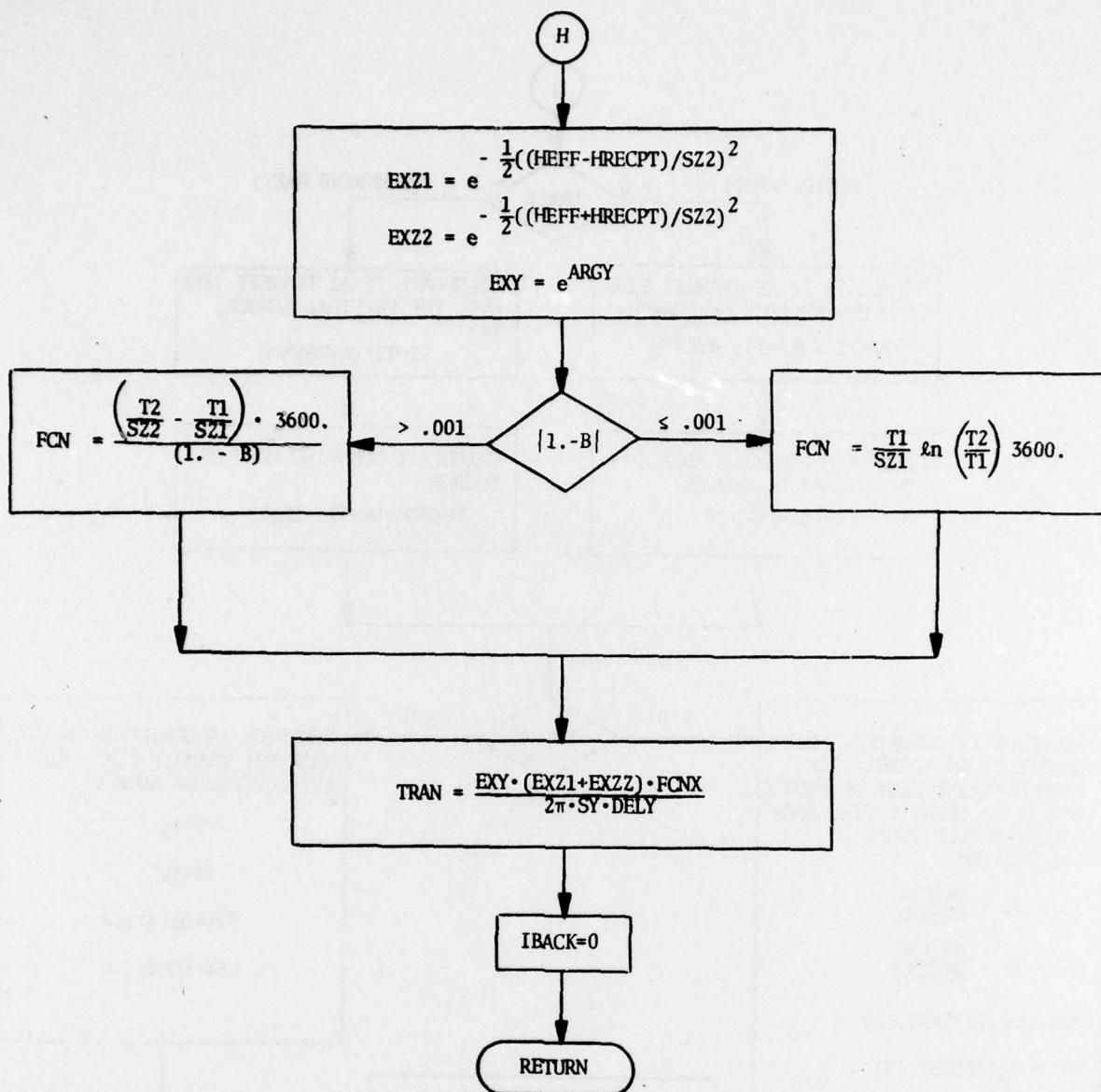














	FUNCTION TRAN (KSTAB,HEFF,NRFLAG,IBACK)	TRAN0000
C		TRAN0001
C	THIS FUNCTION CALCULATES THE COUPLING COEFFICIENT	TRAN0002
C	AT A RECEPTOR DUE TO A POINT OR AREA SOURCE	TRAN0003
C		TRAN0004
	COMMON /INFO/ IRECEP,IWNDIR,ITYPE ,HTAERO,XS,YS,ZS,DELY,DEL7,	TRAN0005
	. TS,VS,DS,HA,PRFLAG,EMIS( 8),NPOL	TRAN0006
	COMMON /MET/ WS,WSMPH,IWS,WD,IWD,SINEWD,COSEWD,JSTAB,HMIX,TEMP,	TRAN0007
	TEMP	TRAN0008
	COMMON /XTRAN/ CL,WSMD,TY,TZ	TRAN0009
	COMMON/WDUH/WSAVE	TRAN0010
	COMMON/LOC/DW,CW,HRECPT,WIT	TRAN0011
	DATA SQ2PI /2.5066283/	TRAN0012
	XL = CL	TRAN0013
	JSTABT=JSTAB	TRAN0014
	WSAVE=WSMD	TRAN0015
	WSKMPH=WSAVE*3.6	TRAN0016
	DELYKM=DELY/1000.	TRAN0017
	ADD1=0.	TRAN0018
	ADD2=0.	TRAN0019
	ADD3=0.	TRAN0020
	ADD4=0.	TRAN0021
	ADD5=0.	TRAN0022
	ADD6=0.	TRAN0023
C		TRAN0024
	IF (KSTAB.GT.0) GO TO 121	TRAN0025
C		TRAN0026
C	KSTAB=0, PLUME IS BELOW THE LID, IF RECEPTOR IS ABOVE	TRAN0027
C	LID, TRAN = 0.	TRAN0028
C		TRAN0029
	IF (HRECPT.GT.HMIX) GO TO 76	TRAN0030
	HLID=HMIX	TRAN0031
	GOTO 140	TRAN0032
C		TRAN0033
C	ASSUME ARBITRARILY HIGH LID HEIGHT FOR TWO CASES WHEN:	TRAN0034
C	KSTAB = 1, PLUME IS INITIALLY ABOVE THE LID	TRAN0035
C	KSTAB = 2, PLUME WILL PENETRATE THE LID	TRAN0036
C	ASSIGN STABILITY CLASSES 5 AND 4 RESPECTIVELY	TRAN0037
C		TRAN0038
	121 JSTABT=6-KSTAB	TRAN0039
	HLID=3050.	TRAN0040
	XL = 100.	TRAN0041
	140 CONTINUE	TRAN0042
	DMIN=ABS(CW)	TRAN0043
	TT=TY+DW/WSKMPH	TRAN0044
	IF (TT.LE.0.) GO TO 76	TRAN0045
	IF (NRFLAG.NE.0) GO TO 143	TRAN0046
C		TRAN0047
C	NRFLAG=0, EFFECTS OF GROUND AND SKY LID ARE TREATED	TRAN0048
C	BY THE MULTIPLE IMAGE METHOD, WITH UP TO 6 TERMS	TRAN0049
C	INCLUDED IN THE COUPLING COEFFICIENT	TRAN0050
C		TRAN0051
	TMIN = 0.	TRAN0052
	IF (DW-WIT.GT.0.) TMIN=(DW-WIT)/WSKMPH	TRAN0053
	IF (IBACK.EQ.0) GO TO 131	TRAN0054
C		TRAN0055
C	IBACK=1, RECEPTOR IS WITHIN CRITICAL DOWNWIND DISTANCE.	TRAN0056
C	TREAT PORTIONS OF SOURCE UPWIND OF CRITICAL LENGTH	TRAN0057
C	FROM RECEPTOR	TRAN0058
C		TRAN0059
	IF (DW+WIT-DELYKM) 132,132,133	TRAN0060
	132 TMIN=0.	TRAN0061

GO TO 131	TRAN0062
133 THIN=(DW+WIT-DELYKH)/WSKNPH	TRAN0063
131 TT=AMAX1 (THIN+TY,.01)	TRAN0064
SY=SIGY (JSTABT,TT)	TRAN0065
XRP = WIT + DW	TRAN0066
IF (XRP.LT.2.*WIT) DW=XRP/2.	TRAN0067
TTT=TZ+DW/WSKNPH	TRAN0068
SZ=SIGZ (JSTABT,TTT)	TRAN0069
ED=DMIN/(SY/1000.)	TRAN0070
EXPD=-.5*ED*ED	TRAN0071
IF (EXPD.LT.-20.) GO TO 76	TRAN0072
EDD=EXP (EXPD)	TRAN0073
IF (DW.GT.2.*XL) GO TO 153	TRAN0074
C	TRAN0075
C CALCULATE MULTIPLE IMAGES	TRAN0076
C	TRAN0077
DENOM=6.2831853*WSAVE*SY*SZ	TRAN0078
C	TRAN0079
C IMAGES 1 AND 2: GROUND REFLECTION OF SOURCE	TRAN0080
C	TRAN0081
Z2=HEFF+HRECPT	TRAN0082
Z1=HEFF-HRECPT	TRAN0083
EXPZ=-(Z1/SZ)**2/2.	TRAN0084
IF (EXPZ.LT.-20.) GO TO 76	TRAN0085
E1=EXP (EXPZ)	TRAN0086
ADD1=E1*EDD/DENOM	TRAN0087
IF (HEFF.GT.0.0) GO TO 171	TRAN0088
ADD2=ADD1	TRAN0089
GOTO 172	TRAN0090
171 CONTINUE	TRAN0091
EXPZ=-(Z2/SZ)**2/2.	TRAN0092
IF (EXPZ.LT.-20.) GO TO 61	TRAN0093
E2=EXP (EXPZ)	TRAN0094
ADD2=ADD1+E2/E1	TRAN0095
C	TRAN0096
C IMAGES 3 AND 4: REFLECTION ABOUT HLID OF SOURCE	TRAN0097
C	TRAN0098
172 CONTINUE	TRAN0099
IF (HEFF.LE.HLID/2.0.AND.SZ.LT.(HLID-HEFF)/2.2) GO TO 61	TRAN0100
IF (HEFF.LT.HLID) GO TO 174	TRAN0101
ADD3=ADD1	TRAN0102
ADD4=ADD2	TRAN0103
GOTO 173	TRAN0104
174 CONTINUE	TRAN0105
Z=2.*HLID-Z2	TRAN0106
EXPZ=-(Z/SZ)**2/2.	TRAN0107
IF (EXPZ.LT.-20.) GO TO 61	TRAN0108
E3=EXP (EXPZ)	TRAN0109
ADD3=ADD1+E3/E1	TRAN0110
Z=2.*HLID-Z1	TRAN0111
EXPZ=-(Z/SZ)**2/2.	TRAN0112
IF (EXPZ.LT.-20.) GO TO 61	TRAN0113
E4=EXP (EXPZ)	TRAN0114
ADD4=ADD1+E4/E1	TRAN0115
IF (HEFF.GT.0.0) GO TO 173	TRAN0116
ADD5=ADD3	TRAN0117
ADD6=ADD4	TRAN0118
GO TO 61	TRAN0119
C	TRAN0120
C IMAGES 5 AND 6: REFLECTION ABOUT HLID OF FIRST BELOW GROUND IMAGE	TRAN0121
C	TRAN0122
173 CONTINUE	TRAN0123



Z=2.*HLID+Z1	TRAN0124
EXPZ=-(Z/SZ)**2/2.	TRAN0125
IF (EXPZ.LT.-20.) GO TO 61	TRAN0126
E5=EXP(EXPZ)	TRAN0127
ADD5=ADD1+E5/E1	TRAN0128
Z=2.*HLID+Z2	TRAN0129
EXPZ=-(Z/SZ)**2/2.	TRAN0130
IF (EXPZ.LT.-20.) GO TO 61	TRAN0131
E6=EXP(EXPZ)	TRAN0132
ADD6=ADD1+E6/E1	TRAN0133
GO TO 61	TRAN0134
C	TRAN0135
C UNIFORM MIXING ASSUMED	TRAN0136
C	TRAN0137
153 DENOM=SQ2PI*WSAVE*SY	TRAN0138
ADD1=EDD/(DENOM*HLID)	TRAN0139
61 CONTINUE	TRAN0140
TRAN=ADD1+ADD2+ADD3+ADD4+ADD5+ADD6	TRAN0141
RETURN	TRAN0142
C	TRAN0143
76 TRAN=0.	TRAN0144
IBACK=0	TRAN0145
RETURN	TRAN0146
C	TRAN0147
C NRFLAG=1, RECEPTOR IS CLOSE TO SOURCE	TRAN0148
C	TRAN0149
143 IF (IBACK.EQ.1) GO TO 144	TRAN0150
C	TRAN0151
C UPWIND PART OF SOURCE	TRAN0152
C	TRAN0153
T2=(DW+WIT)/WSKMPH+TZ	TRAN0154
DWDEDG=DW-WIT	TRAN0155
GO TO 145	TRAN0156
C	TRAN0157
C DOWNWIND PART OF SOURCE	TRAN0158
C	TRAN0159
144 T2=TZ+XL/WSKMPH	TRAN0160
DWDEDG=DW+WIT-DELYKM	TRAN0161
145 IF (DWDEDG.LE.0.) GO TO 146	TRAN0162
C	TRAN0163
C RECEPTOR IS DOWNWIND . FIND TRANSIT TIMES FROM	TRAN0164
C DOWNWIND EDGE OF SOURCE	TRAN0165
C	TRAN0166
T1=TZ+DWDEDG/WSKMPH	TRAN0167
TT=TY+DWDEDG/WSKMPH	TRAN0168
Z1=SIGZ(JSTABT,T1)	TRAN0169
SY=SIGY(JSTABT,TT)	TRAN0170
GO TO 147	TRAN0171
C	TRAN0172
C RECEPTOR IS UPWIND. FIND SHORTEST TRANSIT TIME FROM	TRAN0173
C ANY PORTION OF SOURCE	TRAN0174
C	TRAN0175
146 T1=TZ	TRAN0176
TT=TY	TRAN0177
SZ1=DELZ/2.4	TRAN0178
SY=DELY/2.4	TRAN0179
C	TRAN0180
C COMPUTE COUPLING COEFFICIENT USING INTEGRATED	TRAN0181
C EXPRESSION FOR 'NEAR' SOURCE	TRAN0182
C	TRAN0183
147 SZ2=SIGZ(JSTABT,T2)	TRAN0184
155 ARGY=-(CW*1000./SY)**2/2.	TRAN0185



```

IF (ARGY.LT.-20.) GO TO 76
B=ALOG (SZ1/SZ2)/ALOG (T1/T2)
EXZ1=EXP(-(HEFF-HRECPT)/SZ2)**2/2.)
EXZ2=EXP(-(HEFF+HRECPT)/SZ2)**2/2.)
EY=EXP (ARGY)
IF (ABS(1.-B).LE..001) GO TO 2
FCMX=(T2/SZ2-T1/SZ1)*3600./(1.-B)
GO TO 3
C
C SPECIAL CASE FOR B = 1.
C
C 2 FCMX=T1/SZ1*ALOG (T2/T1) *3600.
C
C 3 TRAN=EY*(EXZ1+EXZ2)*FCMX/(6.2831853*SY*DELY)
  IBACK=0
  RETURN
  END

```

```

TRAN0186
TRAN0187
TRAN0188
TRAN0189
TRAN0190
TRAN0191
TRAN0192
TRAN0193
TRAN0194
TRAN0195
TRAN0196
TRAN0197
TRAN0198
TRAN0199
TRAN0200
TRAN0201
TRAN0202

```

#### REFERENCES

1. Bingaman, D. J. and L. E. Wangen, "Air Quality Assessment Model for Air Force Operations - Source Emissions Inventory Computer Code Documentation." Air Force Civil and Environmental Engineering Development Office report number, CEEDO-TR-76-33.
2. Menicucci, D. F., "Air Quality Assessment Model (AQAM) Data Reduction and Operations Guide," Air Force Weapons Laboratory report number AFWL-75-307, Oct 76.
3. Rote, Donald M., and L. E. Wangen, "A Generalized Air Quality Assessment Model for Air Force Operations - Technical Report," Air Force Weapons Laboratory report number AFWL-TR-74-304, February 1975.

# INITIAL DISTRIBUTION

USAF/PREVP	3
TAC/DEEV	2
CINCUSAFE/Surgeon	1
USAFA/DEV	1
AFIT/DEM	1
OEHL/CC	1
OEHL/OL-AA	5
ADTC/CSV	1
DDC/TCA	12
USA Environ Hygn Agency	1
USA CERL	1
WESTDIV (Code 09BE)	1
Technology Transfer/EPA	1
R&D/EPA	1
NARF/Code 64270	2
Nav Post Grad Sch	2
EESD/DR	5
US Env Prt Agency	2
FAA/AEQ	1
Det 1 HQ ADTC/WE	2
Det 1 HQ ADTC/EC	1
Det 1 HQ ADTC/ECA	7
AFCEC/DEE	3
Dr Delaney	1
AUL/AUL-LSE-70-239	1
Det 1 HQ ADTC/PRL	2
AFATL/DLODL	1
AFATL/DLODR	1